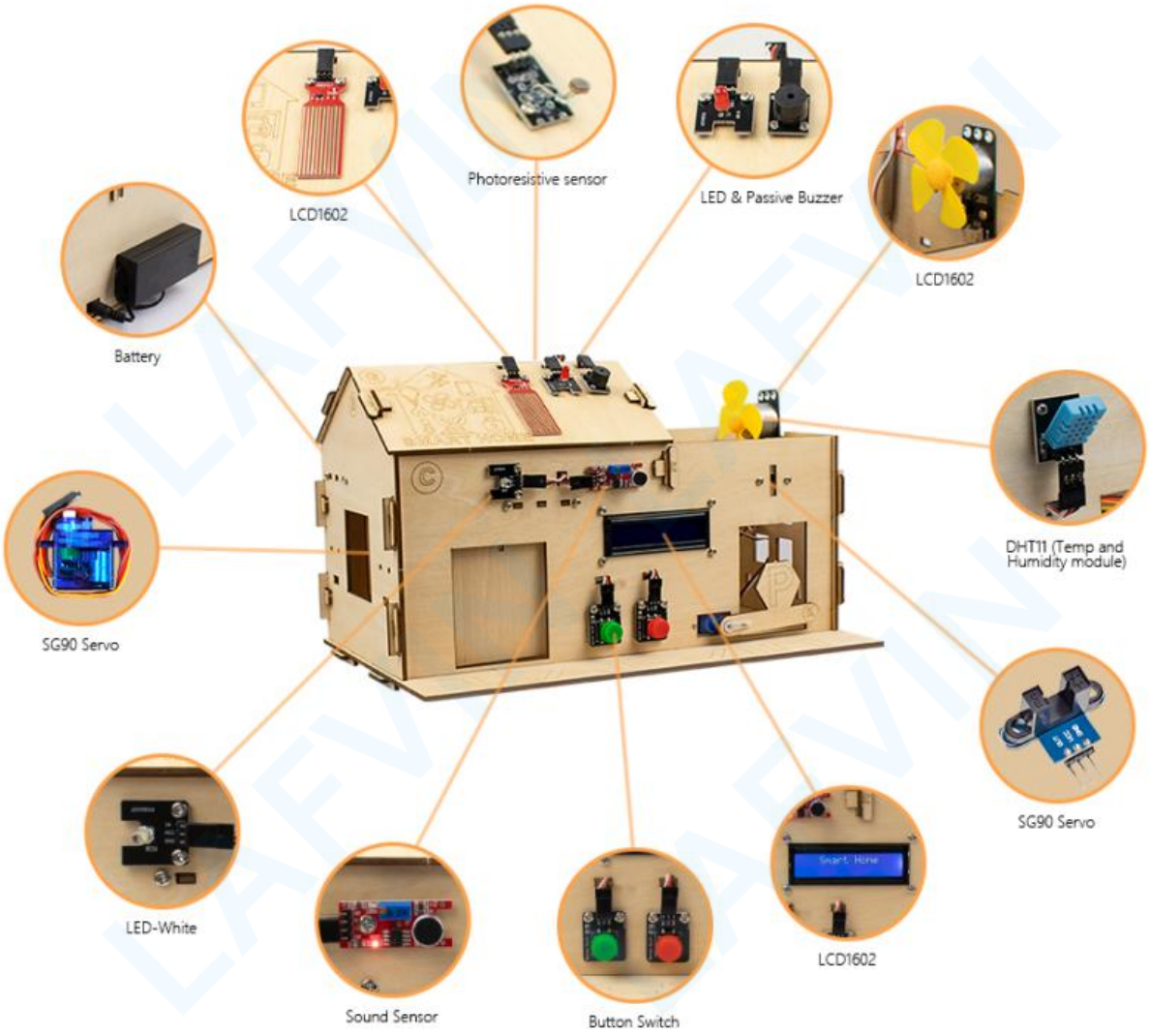
Smart Home Learning Kit


LCD1602
Photoresistive sensor
LED & Passive Buzzer
LCD1602
Battery
DHT11 (Temp and Humidity module)
SG90 Servo
SG90 Servo
LED-White
Sound Sensor
Button Switch
LCD1602

Content

Company Profile

Established in 2011, lafvin is a manufacturer and trader specialized in research,development and production of 2560 uno,nano boards,and all kinds of accessories or sensors use for arduino,rasperrry.We also complete starter kits designed for interested lovers of any levels to learn Arduino or Raspberry.We are located in Shenzhen,China.All of our products comply with international quality standards and are greatly appreciated in a variety of different markets throughout the world.

Customer Service

We are cooperating with a lot of companies from diffirent countries.Also help them to purchase electronic component products in china,and became the biggest supplier of them. We look forward to build cooperate with more companies in future.
By the way,We also look forward to hearing from you and any of your critical comment or suggestions.Pls email us by lafvin_service@163.com if you have any questions or suggestions.As a continuous and fast growing company. We keep striving our best to offer you excellent products and quality service.

Our Store

Aliexpress store: https://www.aliexpress.com/store/1942043 Brand in Amazon:LAFVIN

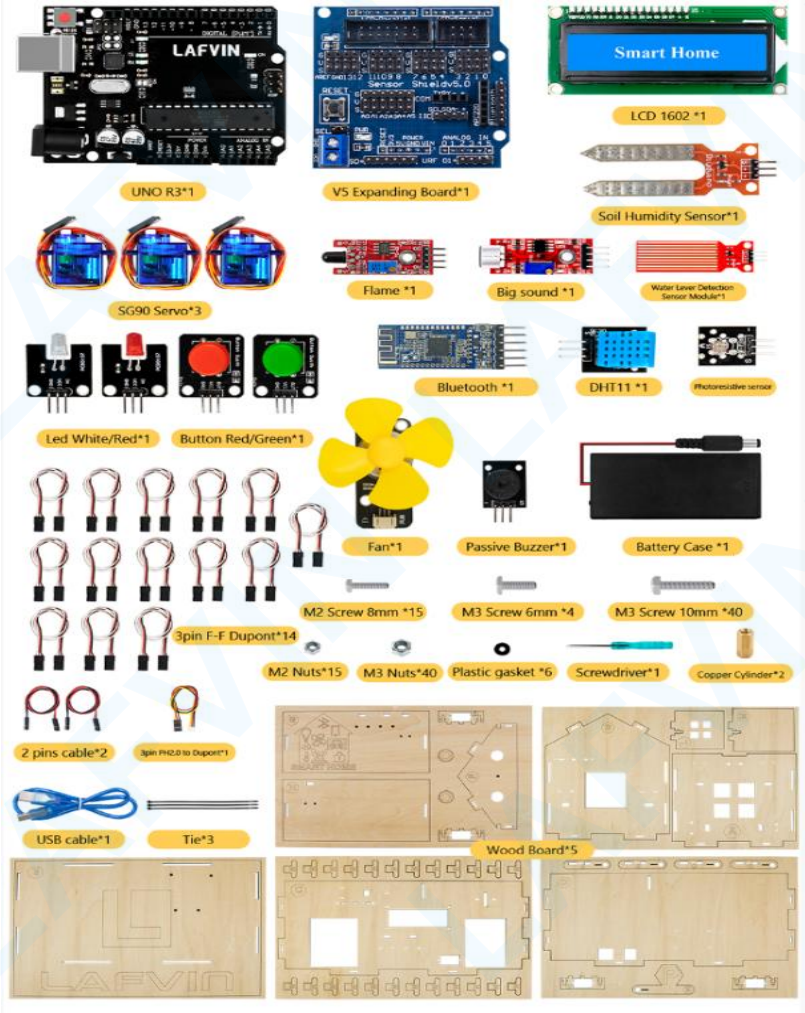
Product Catalog

https://drive.google.com/drive/folders/0BwvEeRN9dKllblZING00TkhYbGs?usp=sharing

Tutorial

LAFVIN Smart Home Learning Kit uses ATmega328P chip as the main controller, with a variety of sensors. Such as soil moisture sensor, DHT11 temperature and humidity sensor, flame detection module and other common environmental detection sensors. The circuit connection of the kit adopts a standardized and easy-to-understand interface, provides 3D dynamic installation video, and supports Arduino IDE and graphical programming at the same time. The APP quickly connects with the low-power Bluetooth module to achieve rich wireless control functions. This learning kit is very useful for learners who are keen to explore electronic knowledge and programming.

Packing List

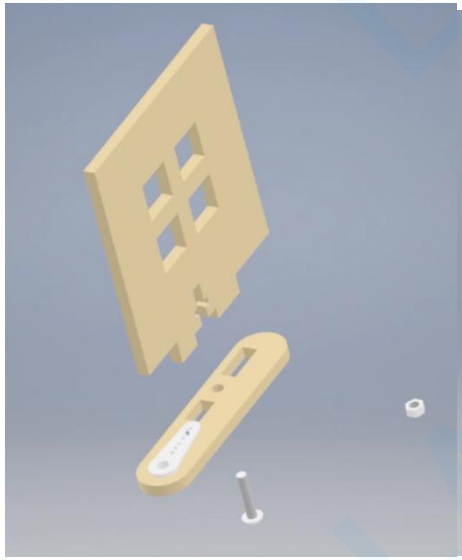


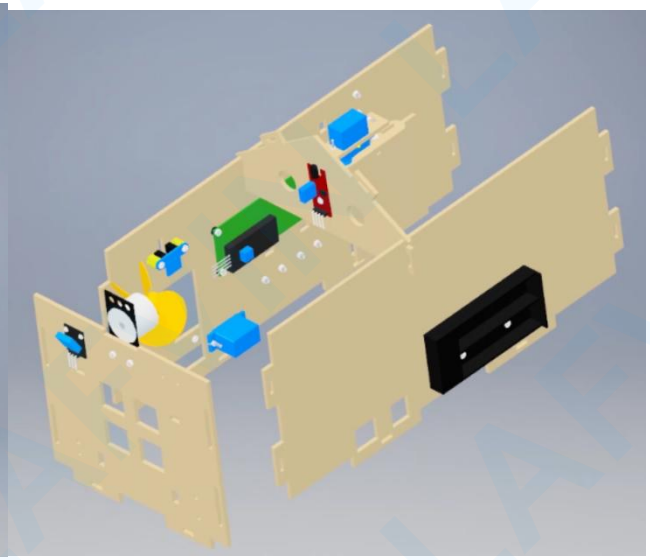
1.Smart Home Learning Kit Introduction

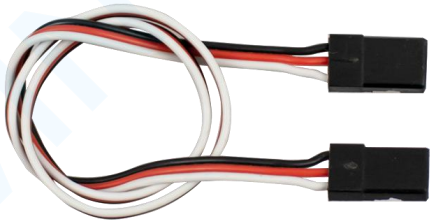
The Smart Home Learning Kit is mainly manufactured using the arduino uno R3 main control and High-quality DIY assembly wood board, and a wealth of sensor module.

Smart Home Learning Kit have the following advantages:

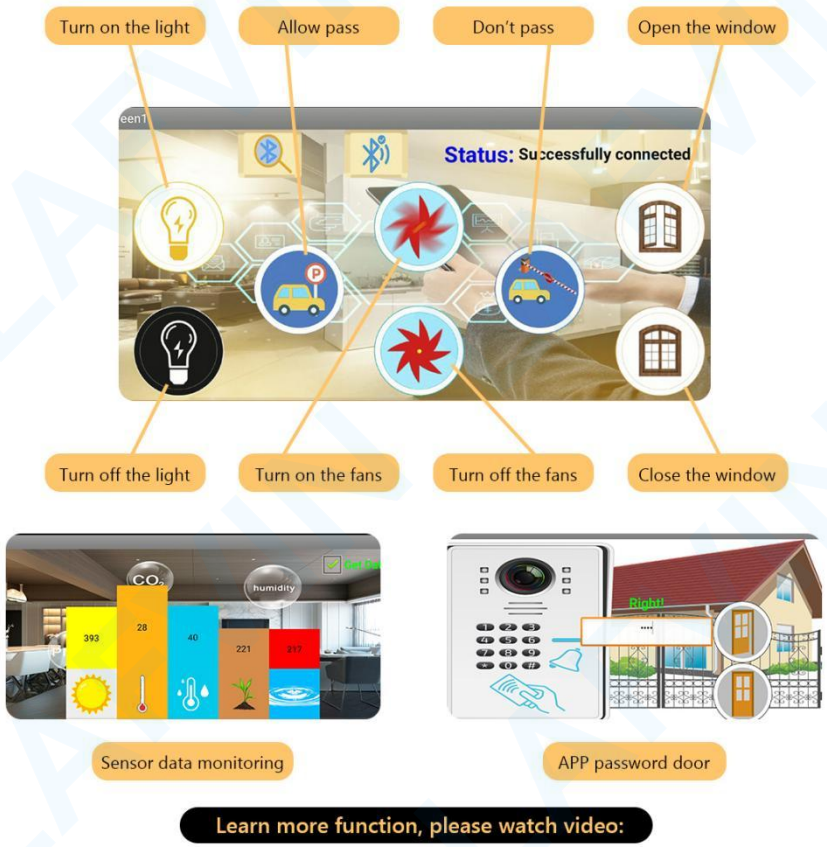
1) Simple and easy-to-understand wooden board assembly structure, And provide 3D dynamic installation video tutorial.the sensor connection line adopts 3pin F-F Dupont line to avoid complicated wiring.



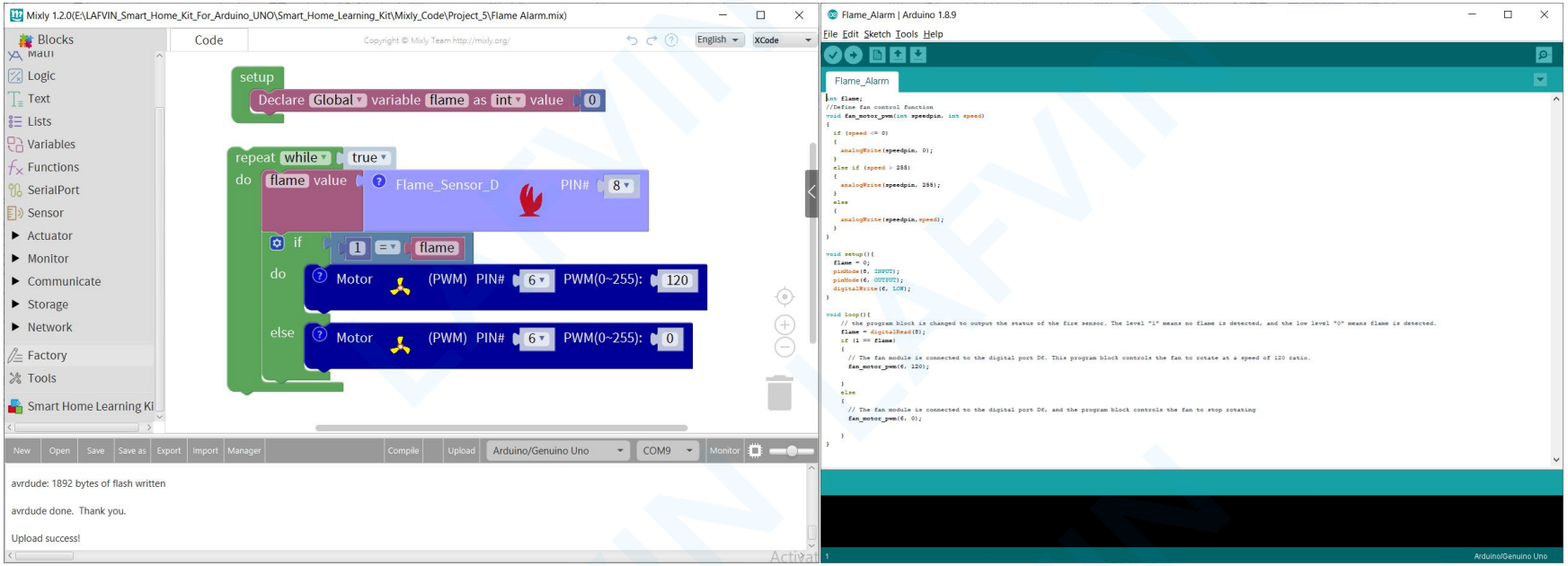




2) The low-power Bluetooth 4.0 module is used to simplify the connection steps with the mobile phone APP, and provide convenience for introductory learning. The APP has powerful functions, including wireless control actuators, sensor data monitoring, and wireless password opening functions.



3)Simultaneously supports mixly graphical programming and arduino IDE code programming, which is convenient for beginners to learn.



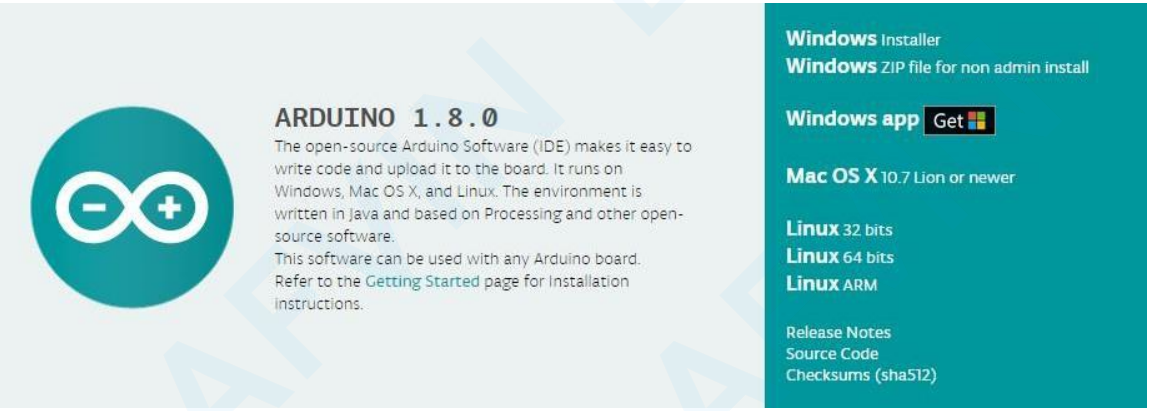
2.Getting Started with Arduino IDE

2.1 How to Install Arduino IDE

Introduction

The Arduino Integrated Development Environment (IDE) is the software side of the Arduino platform.In this Project, you will learn how to setup your computer to use Arduino and how to set about the Projects that follow.The Arduino software that you will use to program your Arduino is available for Windows, Mac and Linux. The installation process is different for all three platforms and unfortunately there is a certain amount of manual work to install the software.

STEP 1: Go to [https://www.arduino.cc/en/Main/Software and find below page.](https://www.arduino.cc/en/Main/Software)

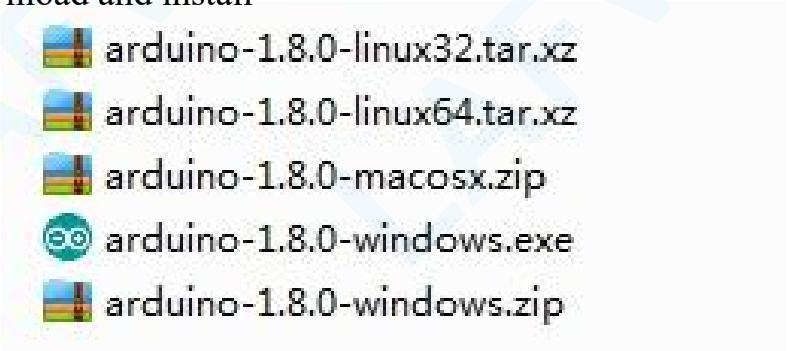


The version available at this website is usually the latest version, and the actual version may be newer than the version in the picture.

STEP2：Download the development software that is compatible with the operating system of your computer.
Take Windows as an example here.

Click Windows Installer.

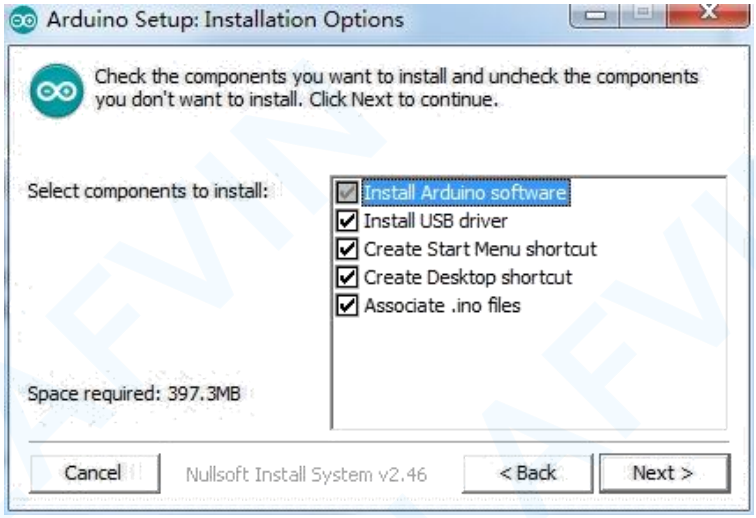
Click JUST DOWNLOAD.
Also version 1.8.0 is available in the material we provided, and the versions of our materials are the latest versions when this course was made.
You can choose the latest version to download and install"



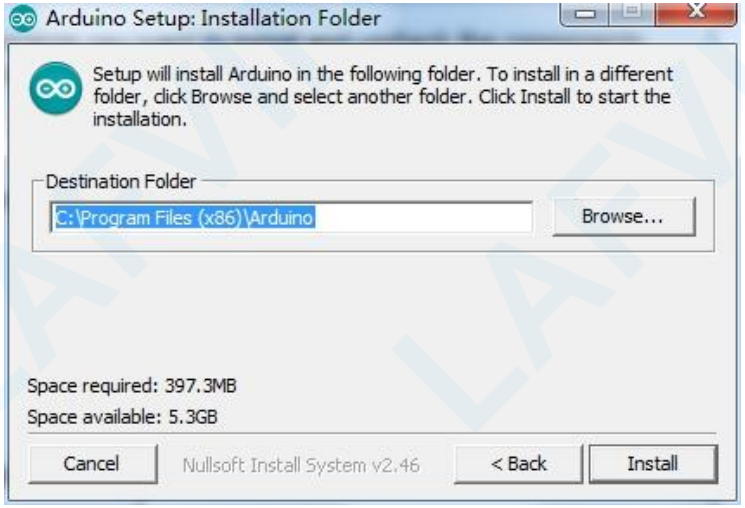
Installing Arduino (Windows)

Install Arduino with the exe. Installation package.

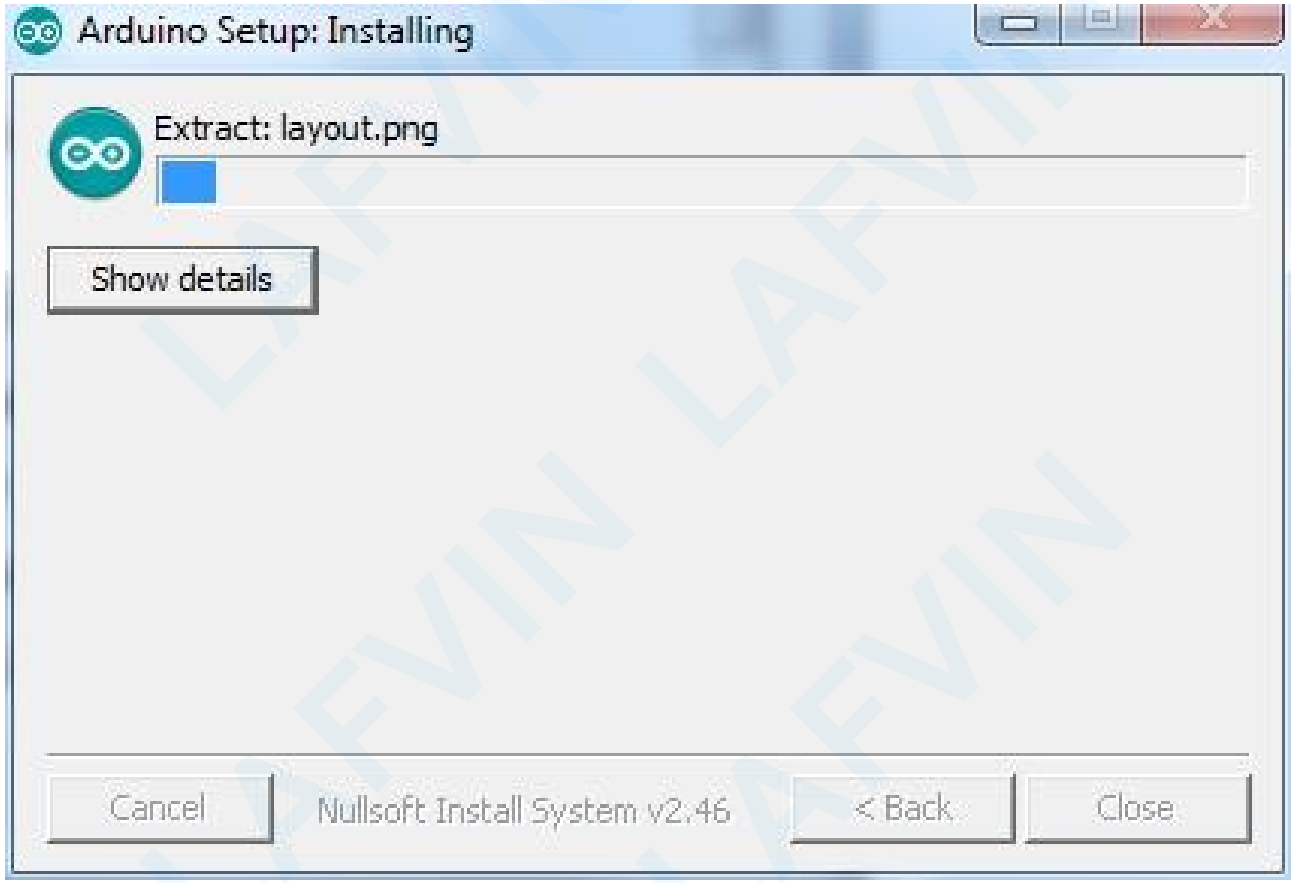
Click I Agree to see the following interface


Arduino Setup: Installation Options
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.
Select components to install:
Install Arduino software
Install USB driver
Create Start Menu shortcut
Create Desktop shortcut
Associate .ino files
Space required: 397.3MB
Cancel
Nullsoft Install System v2.46
< Back
Next >

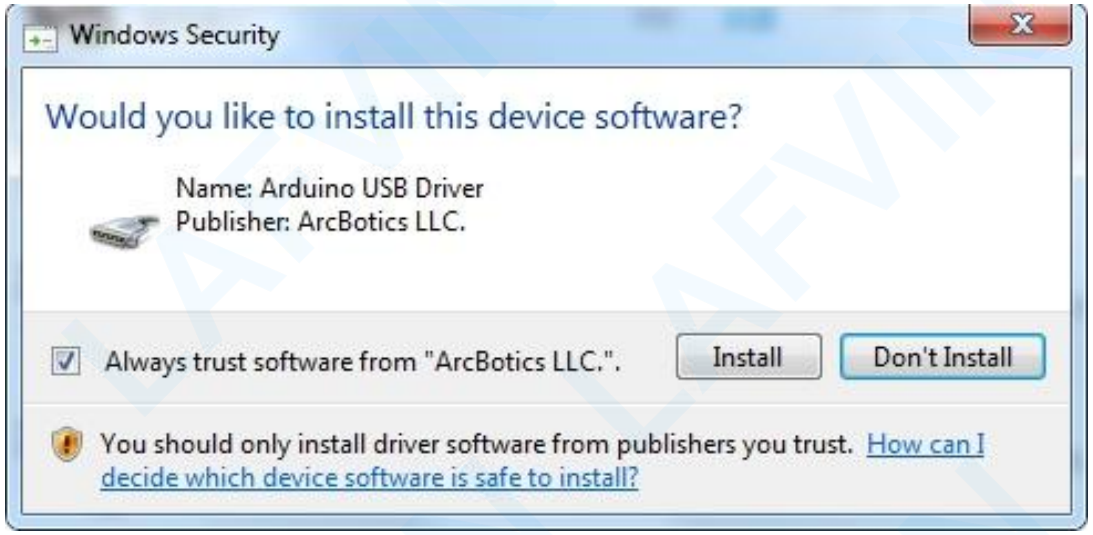
Click Next


Arduino Setup: Installation Folder
Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.
Destination Folder
C:\Program Files (x86)\Arduino
Browse...
Space required: 397.3MB
Space available: 5.3GB
Cancel
Nullsoft Install System v2.46
< Back
Install

You can press Browse… to choose an installation path or directly type in the directory you want.
Click Install to initiate installation

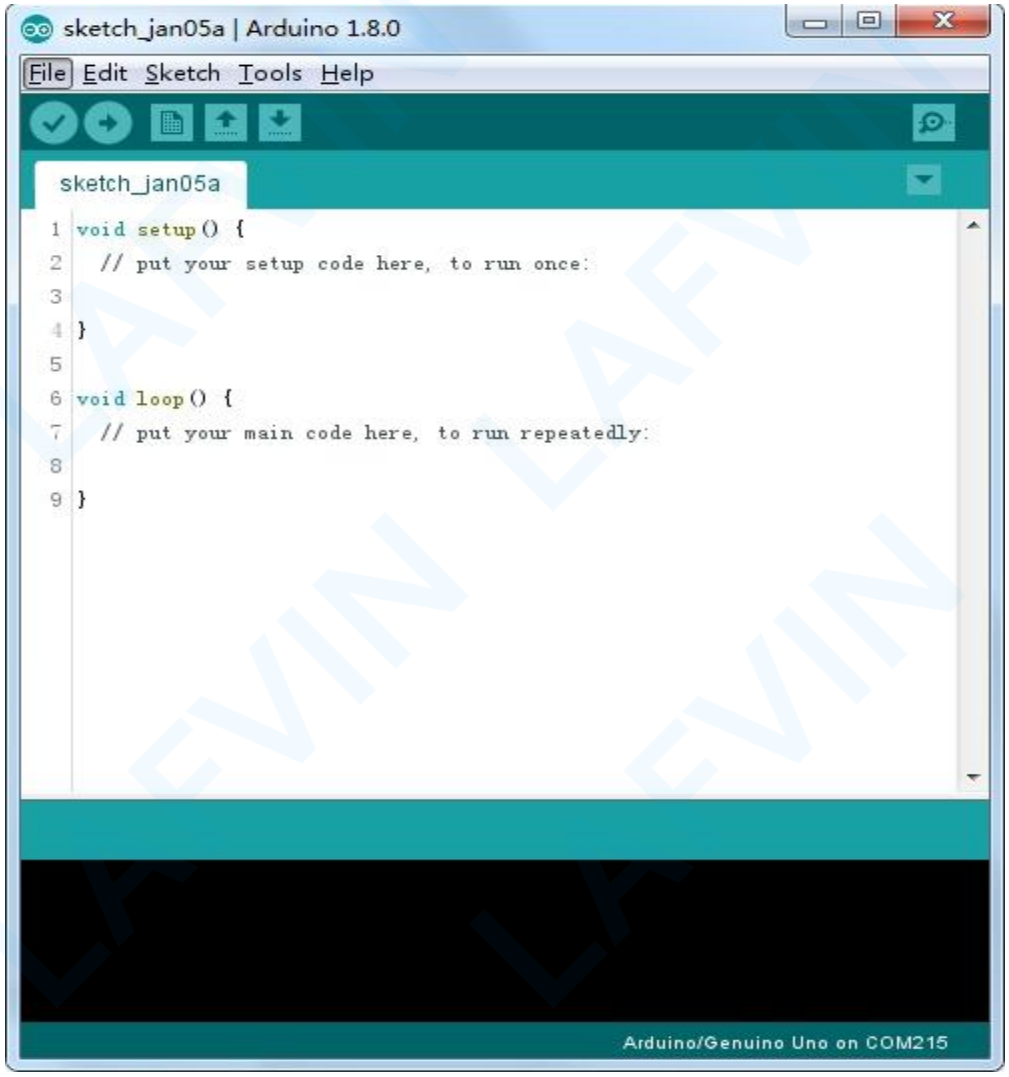


Wait for the installing process, if appear the interface of Window Security, just continue to click Install to finish the installation.



Next, the following icon appears on the desktop

Double-click to enter the desired development environment

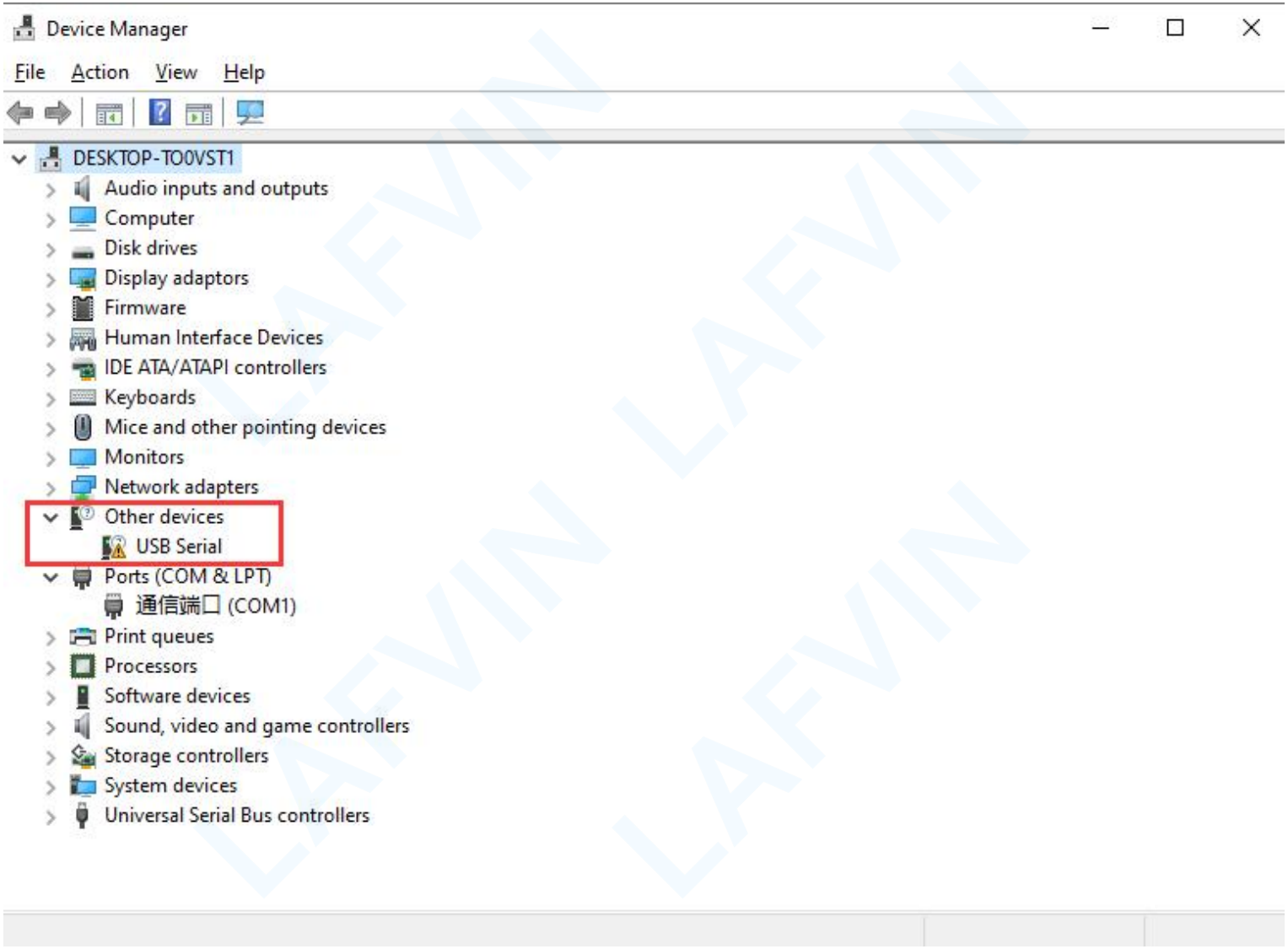


2.2 How to Install Arduino Driver

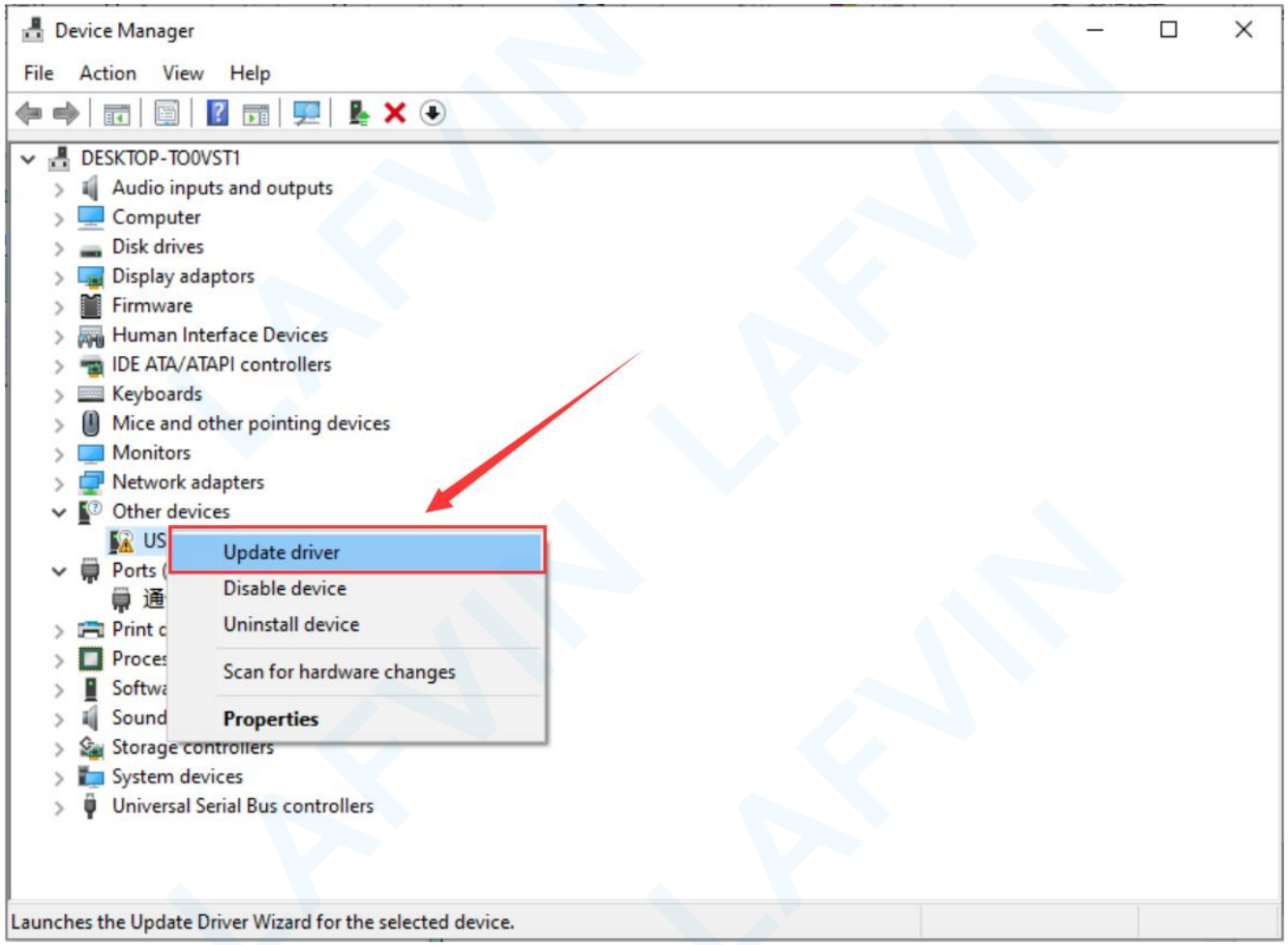
Next, we will introduce the driver installation of UNO R3 development board. The driver installation may have slight differences in different computer systems. So in the following let's move on to the driver installation in the Window system.

The Arduino folder contains both the Arduino program itself and the drivers that allow the Arduino to be connected to your computer by a USB cable. Before we launch the Arduino software, you are going to install the USB drivers.

When you connect UNO board to your computer at the first time, right click the icon of your "Computer" —>for "Properties"—> click the "Device manager" ,
under "Other Devices"or"USB-Serial", you should see an icon for "Unknown device" with a little yellow warning triangle next to it. This is your Arduino.Or you can search for "devi" in your computer, or you can open the device manager of your computer.

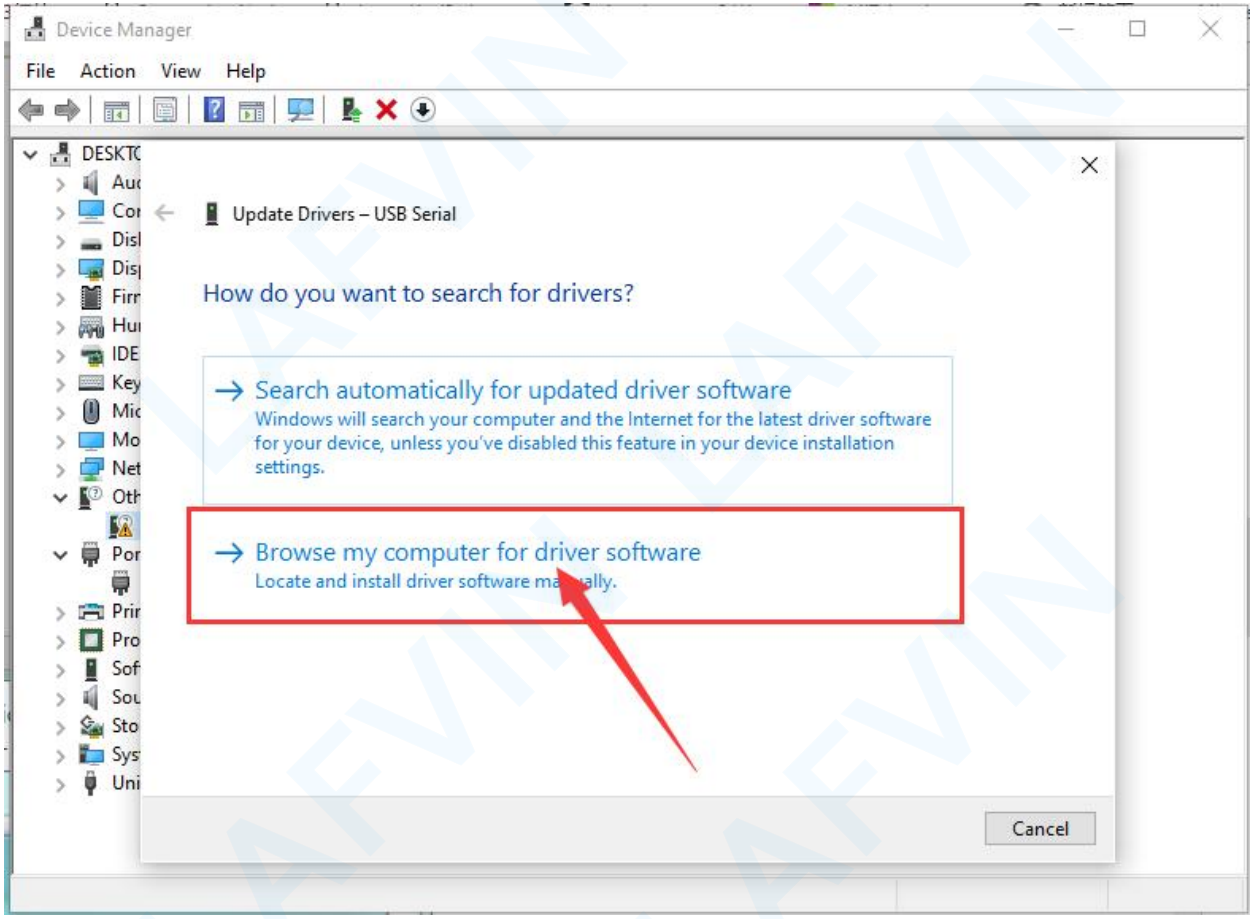

Device Manager
File Action View Help
DESKTOP-TO0VST1
Audio inputs and outputs
Computer
Disk drives
Display adaptors
Firmware
Human Interface Devices
IDE ATA/ATAPI controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Other devices
USB Serial
Ports (COM & LPT)
通信端口 (COM1)
Print queues
Processors
Software devices
Sound, video and game controllers
Storage controllers
System devices
Universal Serial Bus controllers

Then right-click on the device and select the top menu option (Update Driver Software...) shown as the figure below.



Then it will be prompted to either “Search Automatically for updated driver software” or “Browse my computer for driver

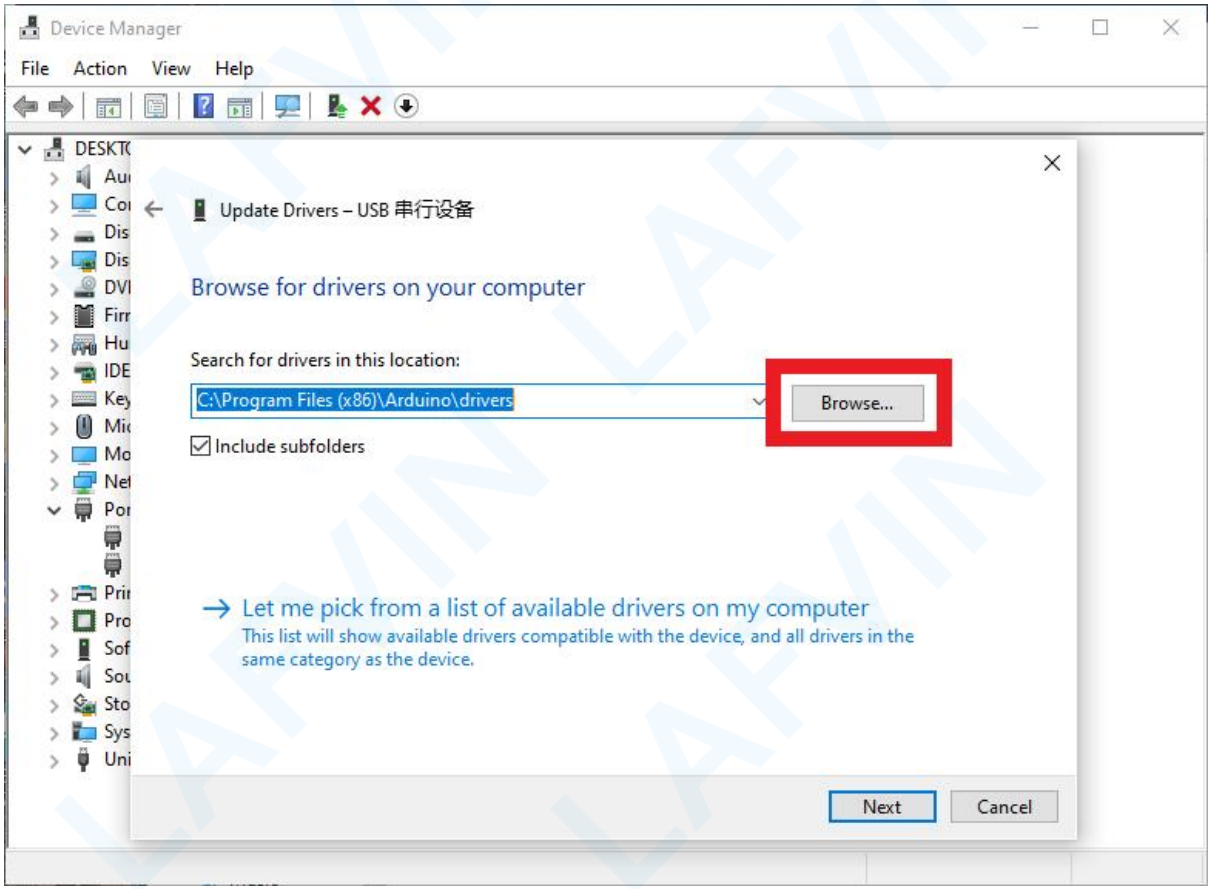
software". Shown as below. In this page, select "Browse my computer for driver software".



Right-click on the device and select the top menu option (Update Driver Software...).

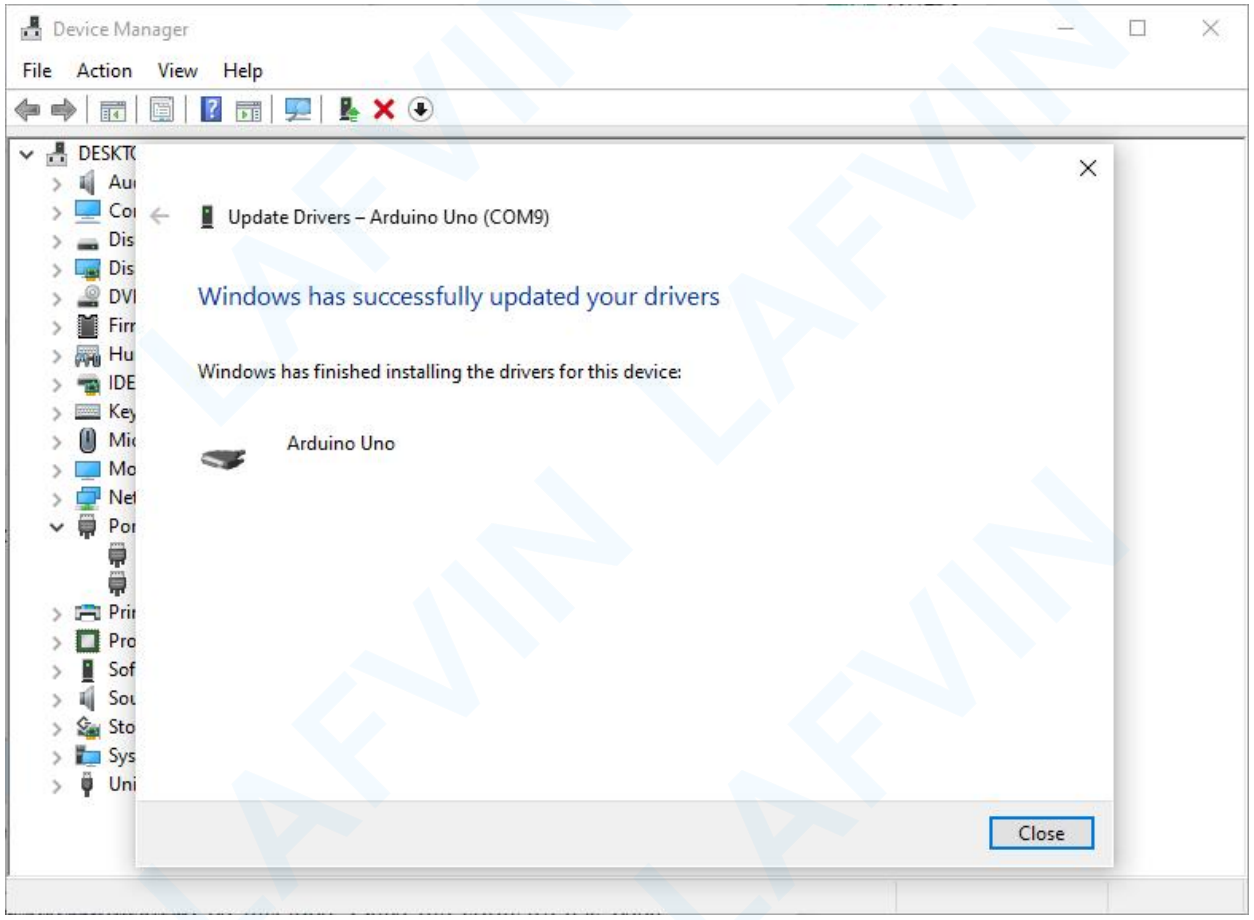
You will then be prompted to either 'Search Automatically for updated driver software' or 'Browse my computer for driver

software’. Select the option to browse and navigate to the :**C\Program Files(x86)\Arduino\drivers.**(Note: Here is the path you choose to install arduino IDE. The path chosen in the installation tutorial in the previous section is that, so the path I chose is**C\Program Files(x86)\Arduino\drivers**)

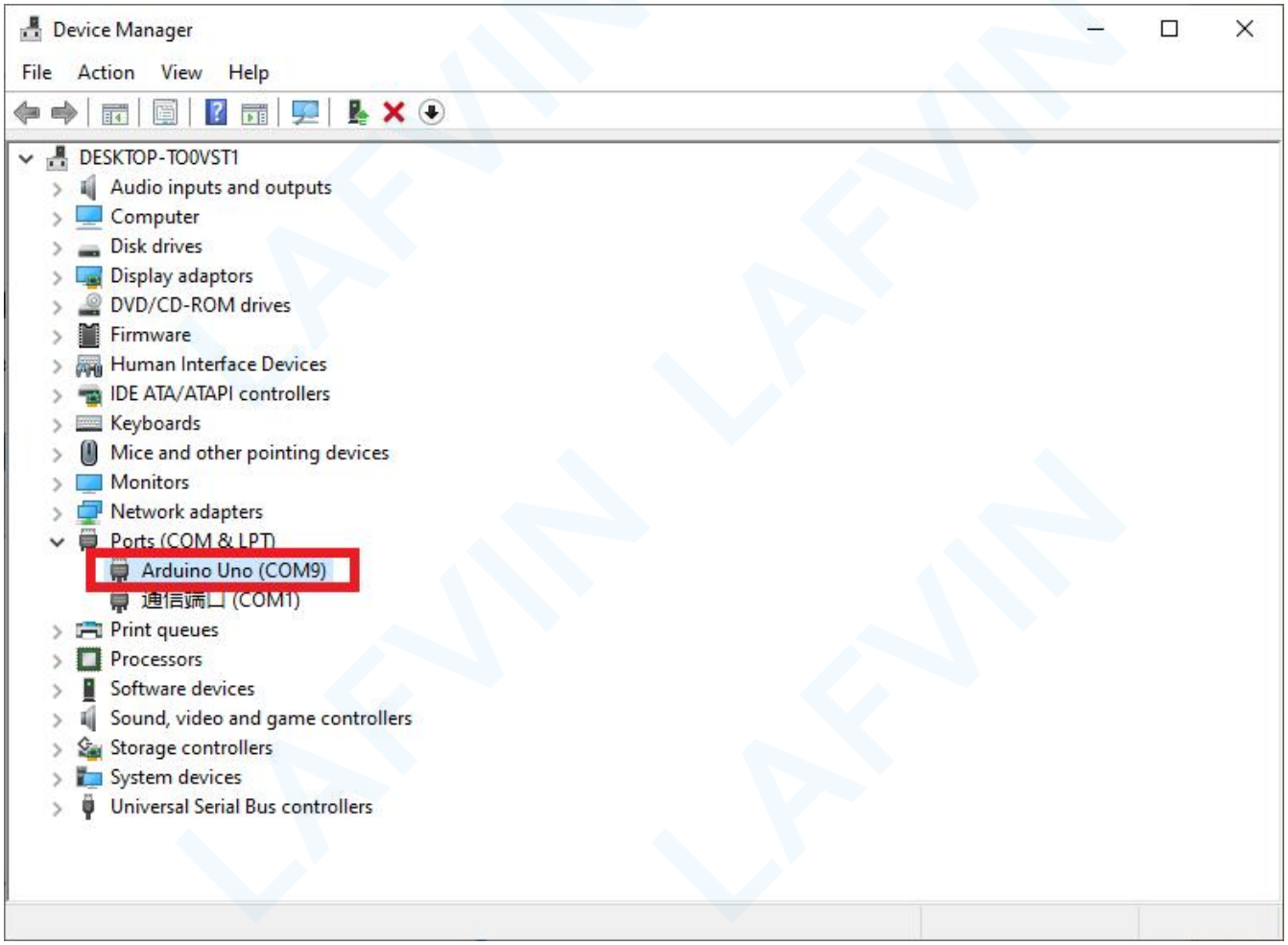


Click “Next” and you may get a security warning, if so, allow the software to be installed.

Once the software has been installed, you will get a confirmation message. Installation completed,click “Close”.



Up to now, the driver is installed well. Then you can right click “Computer”—>“Properties”—>“Device manager” , you should see the device as the figure shown below.



2.3 How to Add Libraries

Installing Additional Arduino Libraries

Once you are comfortable with the Arduino software and using the built-in functions, you may want to extend the ability of your Arduino with additional libraries.

What are Libraries?

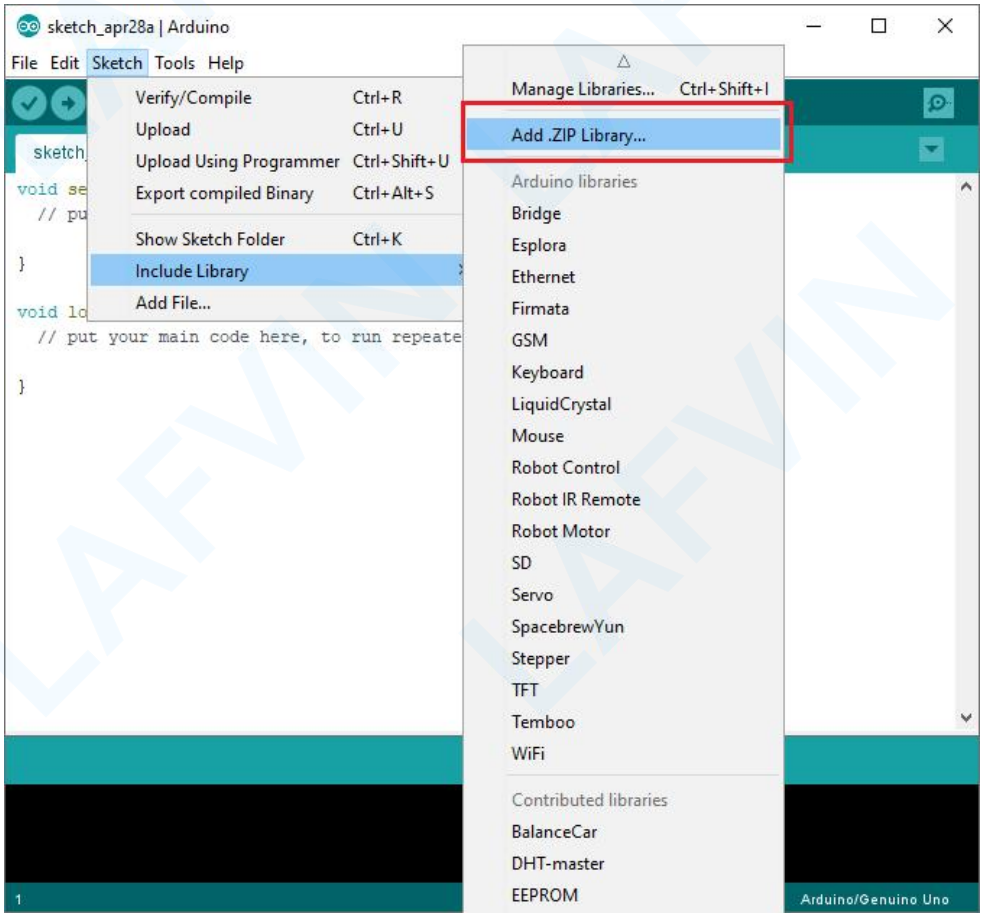
Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc. For example, the built-in LiquidCrystal library makes it easy to talk to character LCD displays. There are hundreds of additional libraries available on the Internet for download. The built-in libraries and some of these additional libraries are listed in the reference. To use the additional libraries, you will need to install them.

How to Install a Library

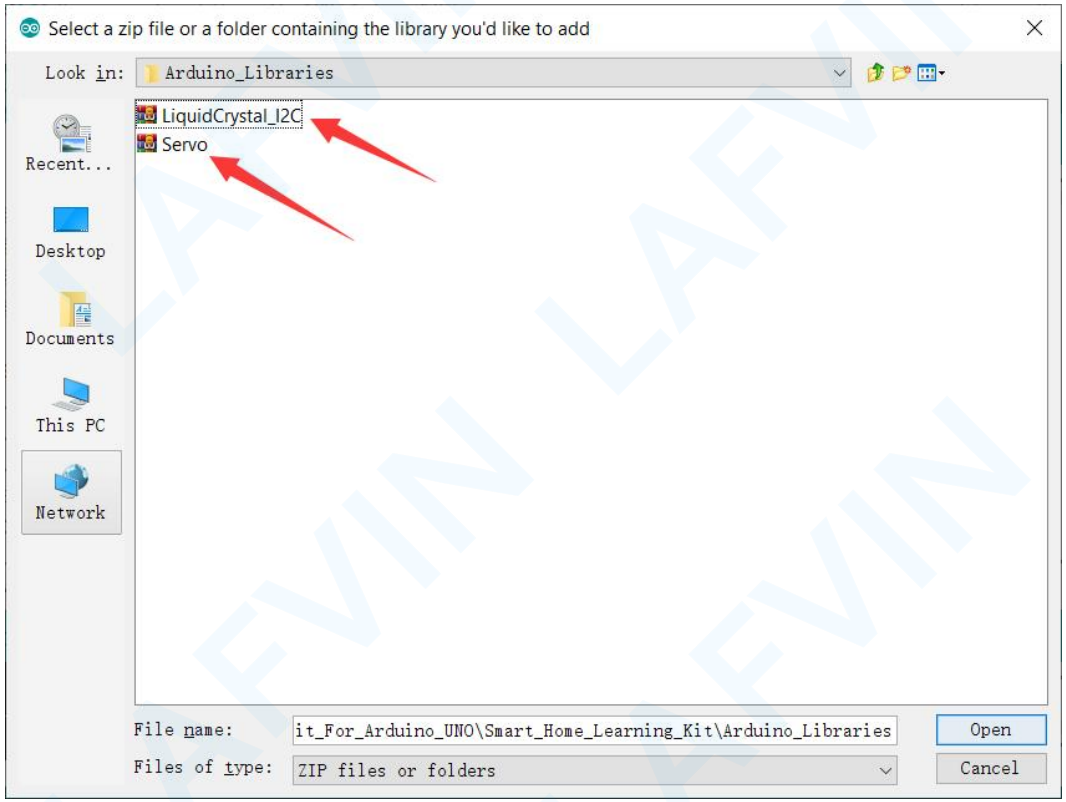
Importing a .zip Library

Libraries are often distributed as a ZIP file or folder. The name of the folder is the name of the library. Inside the folder will be a .cpp file, a .h file and often a keywords.txt file, examples folder, and other files required by the library. you can install 3rd party libraries in the IDE. Do not unzip the downloaded library, leave it as is.

In the Arduino IDE, navigate to Sketch > Include Library. At the top of the drop down list, select the option to "Add .ZIP Library".



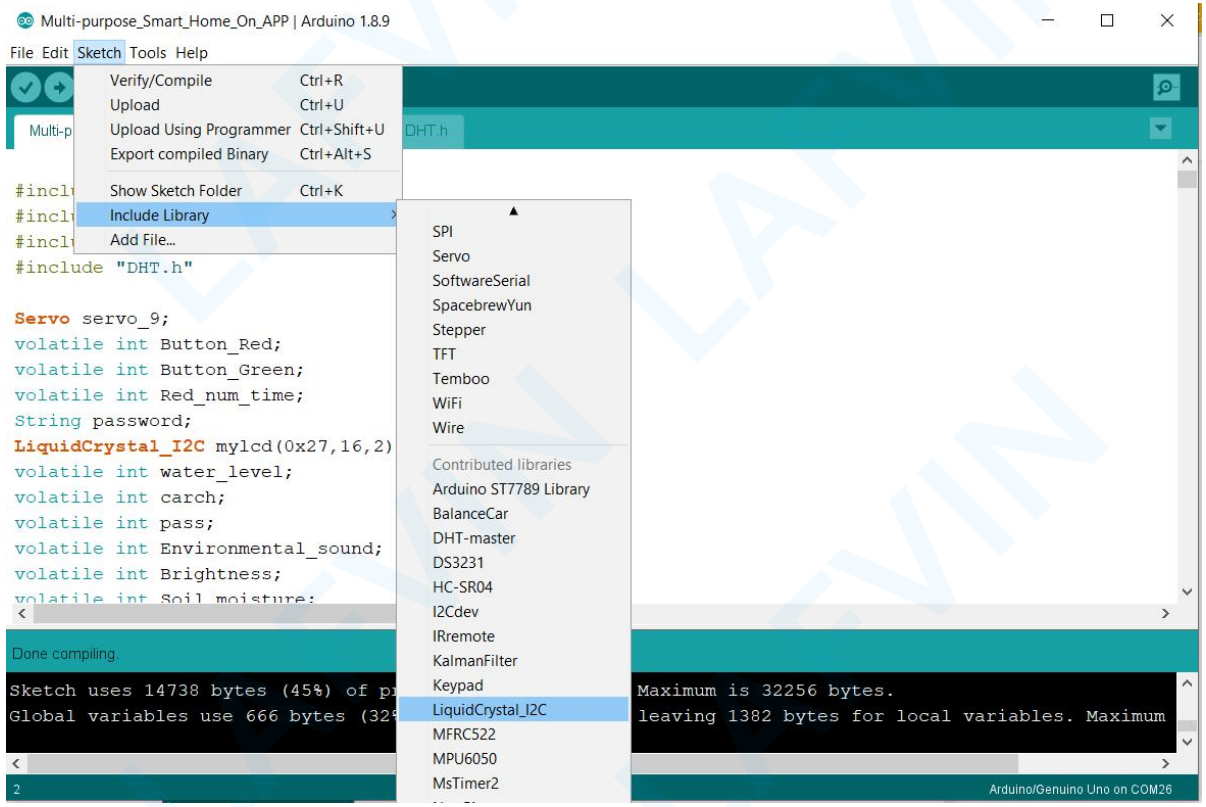
You will be prompted to select the library you would like to add. Navigate to the .zip file's location and open it.
In this learning kit you need to add two library files which are <Servo><LiquidCrystal_I2C>. You can add this library file in the learning files provided by us.



(Note: The compression format of the library file must be .zip, if it is a .rar compression format, the addition will fail.)

Return to the Sketch > Import Library menu. You should now see the library at the bottom of the drop-down menu. It is ready to be used in your sketch. The zip file will have been expanded in the libraries folder in your Arduino sketches directory.

NB: the Library will be available to use in sketches, but examples for the library will not be exposed in the File > Examples until after the IDE has restarted.



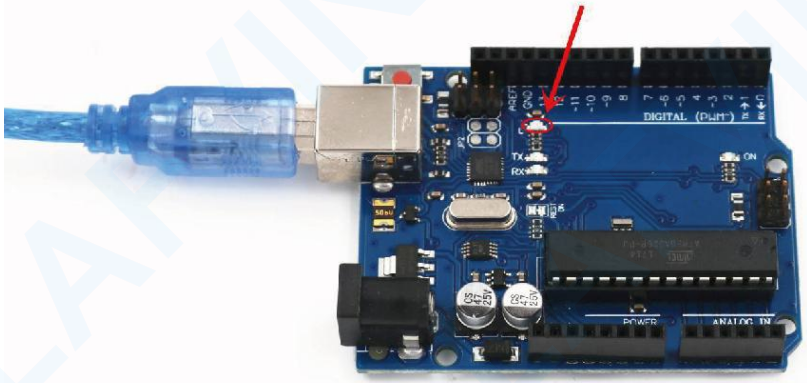
Those two are the most common approaches. MAC and Linux systems can be handled likewise. The manual installation to be introduced below as an alternative may be seldom used and users with no needs may skip it.

2.4 Blink Test

you will learn how to program your UNO controller board to blink the Arduino's built-in LED, and how to download programs by basic steps.

The UNO board has rows of connectors along both sides that are used to connect to several electronic devices and plug-in 'shields' that extends its capability.

It also has a single LED that you can control from your sketches. This LED is built onto the UNO board and is often referred to as the 'L' LED as this is how it is labeled on the board.



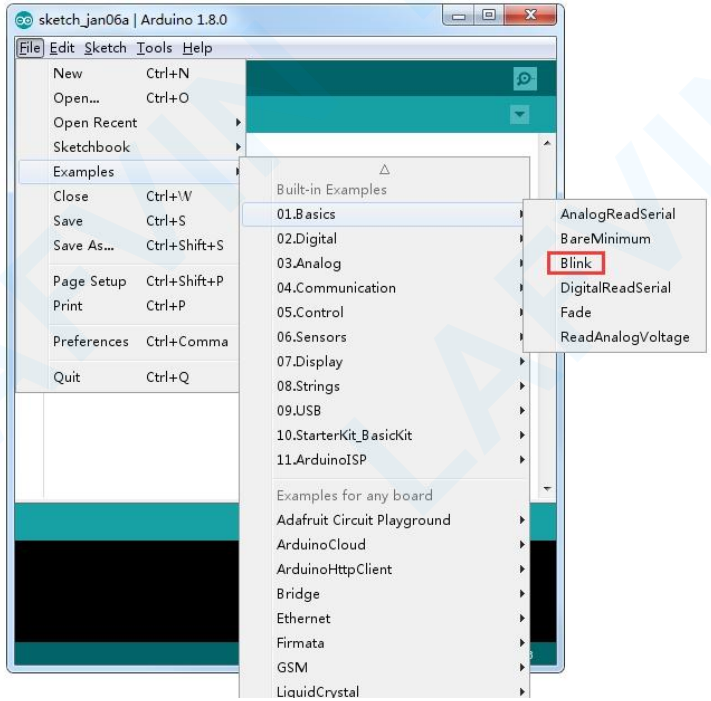
You may find that your UNO board's 'L' LED already blinks when you connect it to a USB plug. This is because the boards are generally shipped with the 'Blink' sketch pre-installed.

In this Project, we will reprogram the UNO board with our own Blink sketch and then change the rate at which it blinks.

In the previous chapter-How to install Arduino IDE, you set up your Arduino IDE and made sure that you could find the right serial port for it to connect to your UNO board. The time has now come to put that connection to the test and program your UNO board.

The Arduino IDE includes a large collection of example sketches that you can load up and use. This includes an example sketch for making the 'L' LED blink.

Load the 'Blink' sketch that you will find in the IDE's menu system under File > Examples > 01.Basics

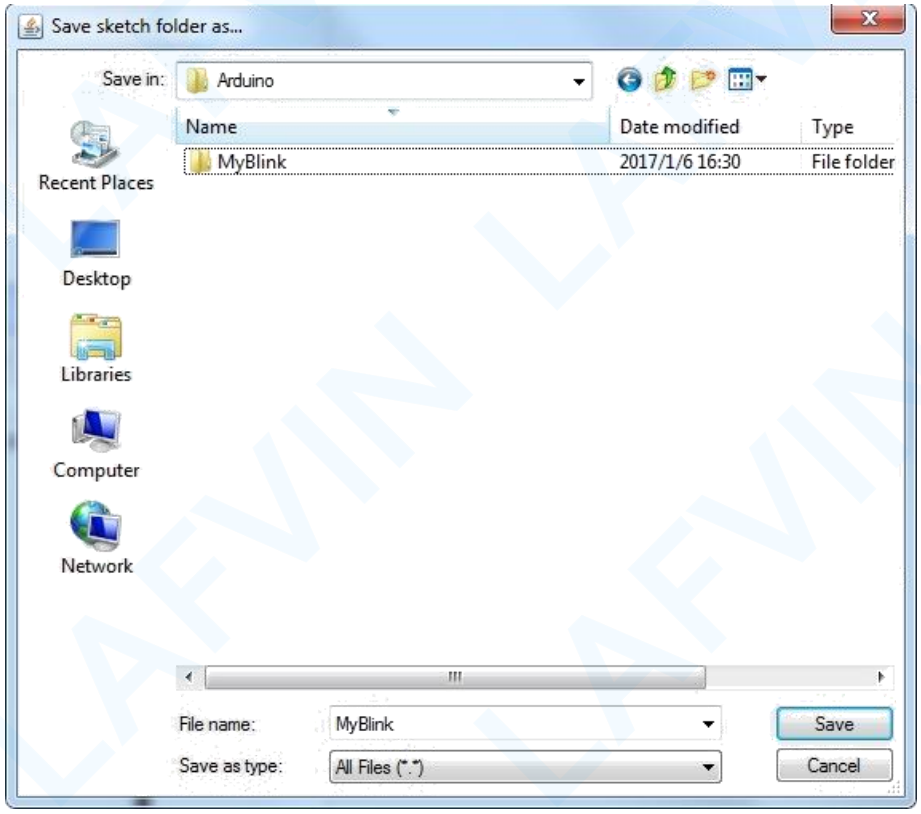


When the sketch window opens, enlarge it so that you can see the entire sketch in the window.

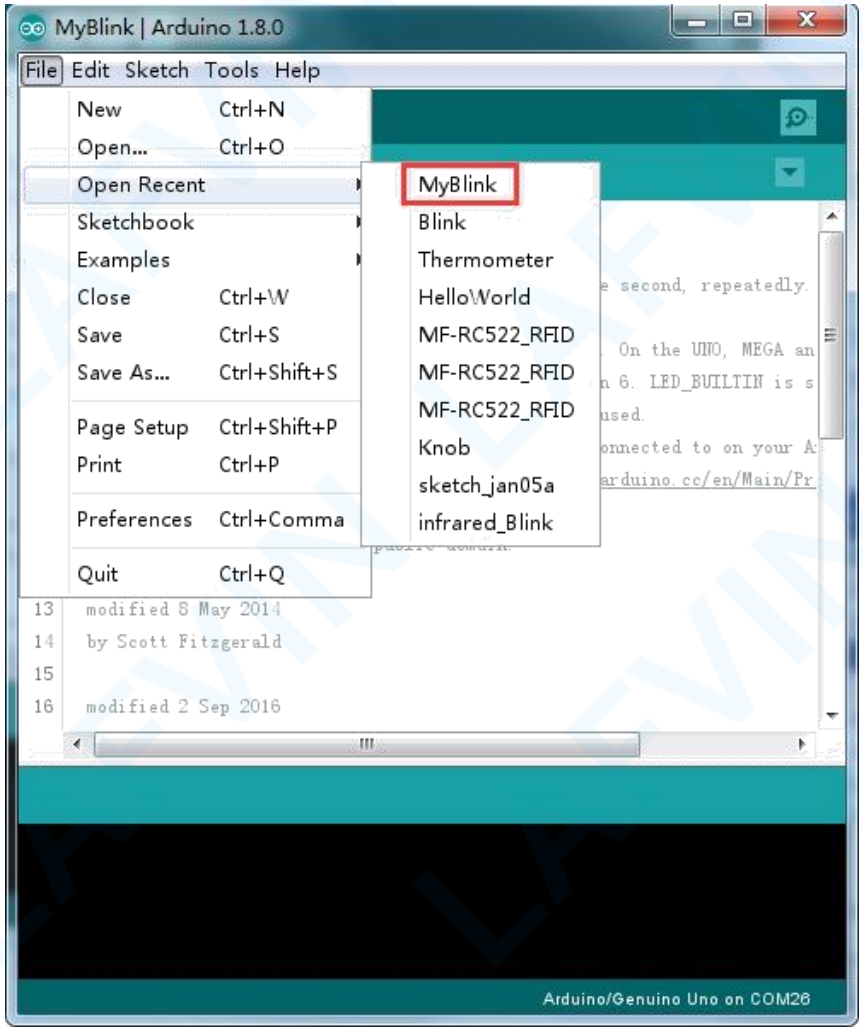
The example sketches included with the Arduino IDE are 'read-only'. That is, you can upload them to an UNO R3 board, but if you change them, you cannot save them as the same file.

Since we are going to change this sketch, the first thing you need to do is save your own copy.

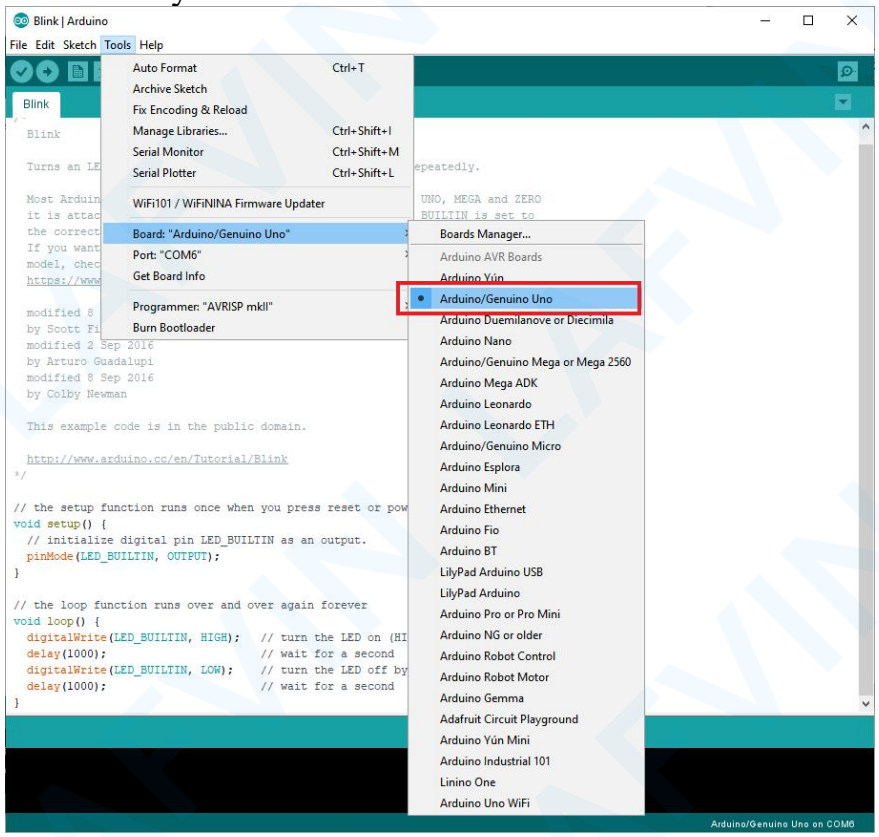
From the File menu on the Arduino IDE, select 'Save As..' and then save the sketch with the name 'MyBlink'.

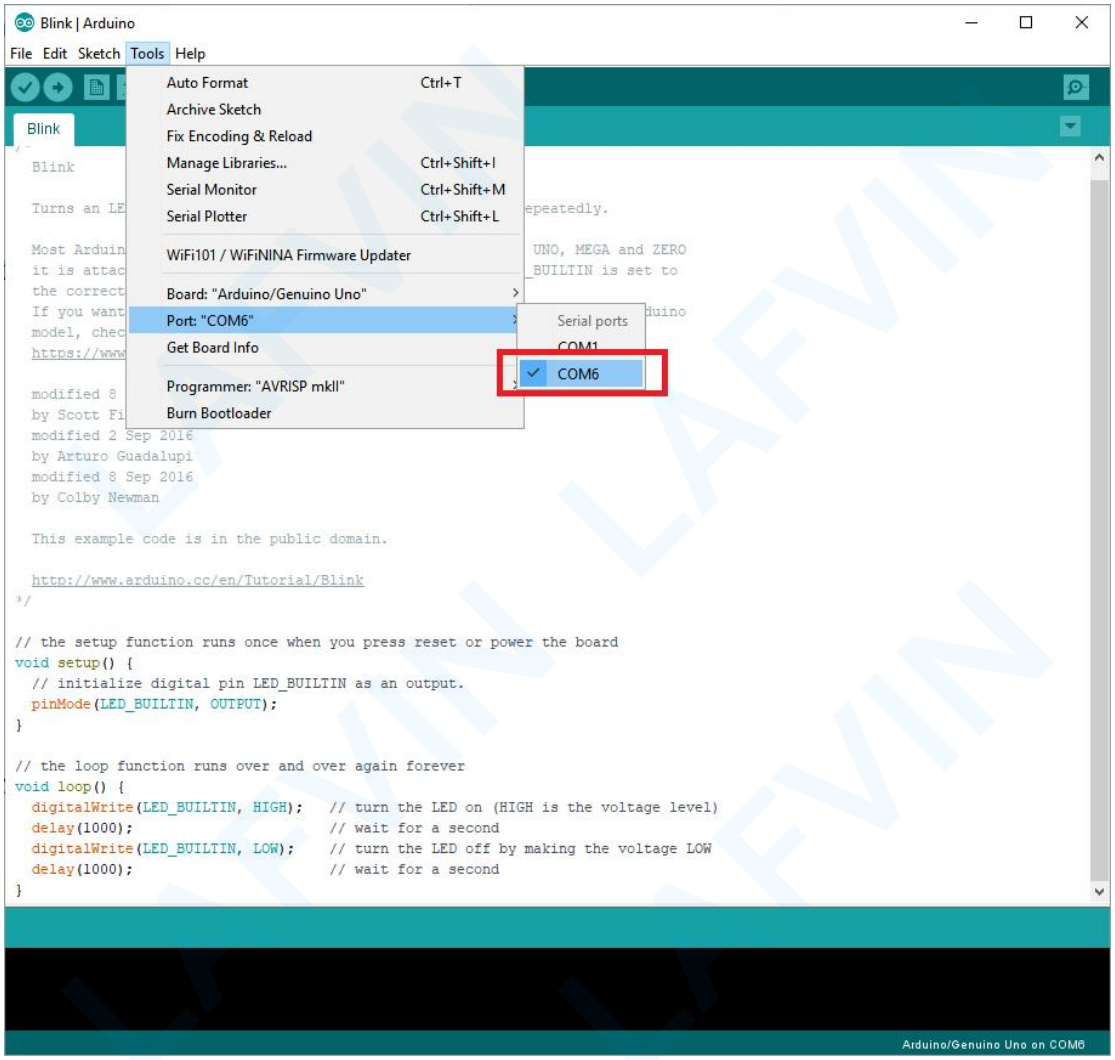


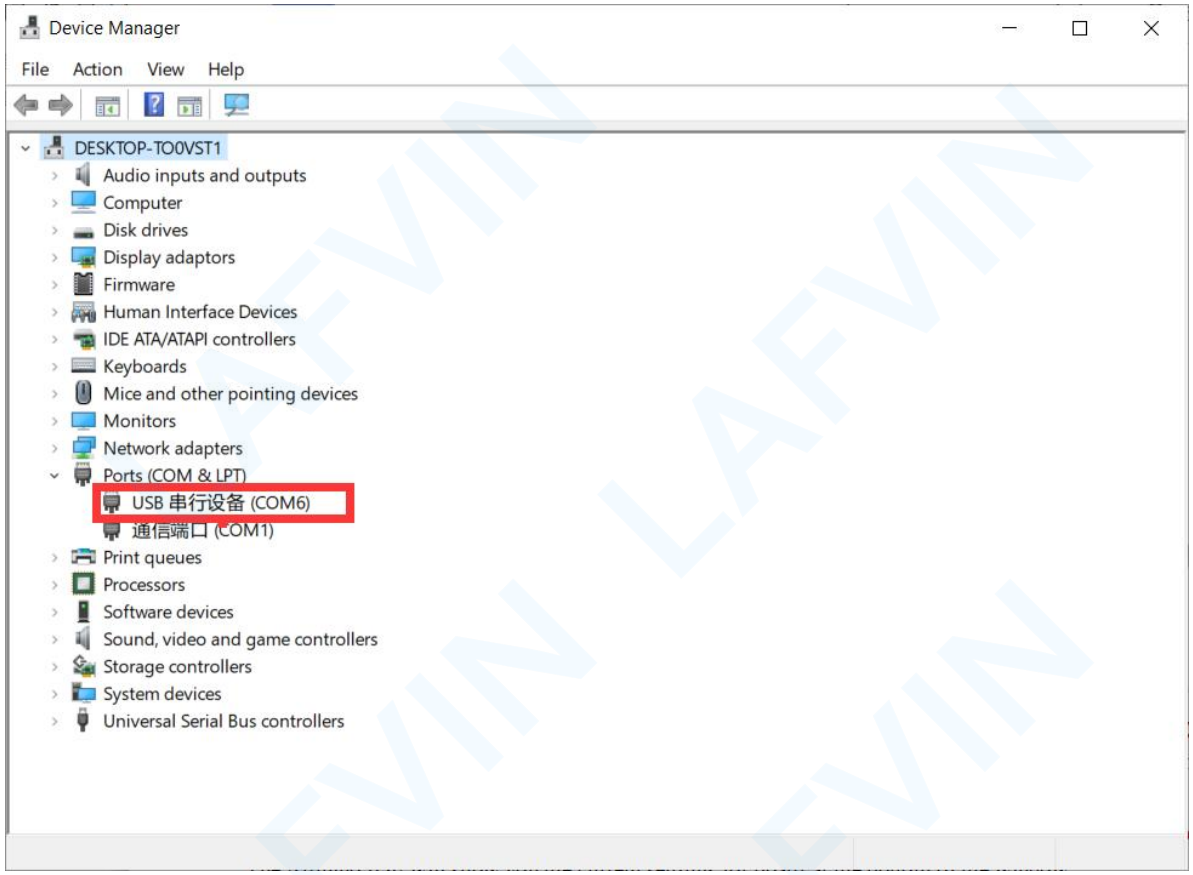
You have saved your copy of 'Blink' in your sketchbook. This means that if you ever want to find it again, you can just open it using the File > Sketchbook menu option.



Attach your Arduino board to your computer with the USB cable and check that the 'Board Type' and 'Serial Port' are set correctly.



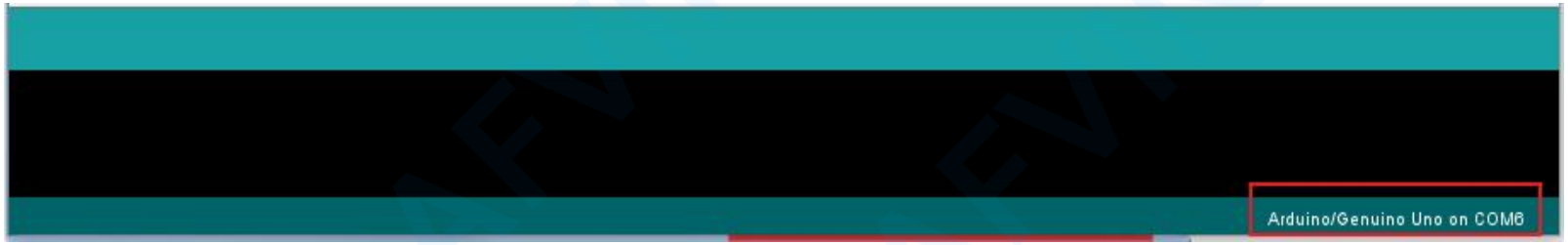

Blink | Arduino
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM6"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
COM1
COM6
Blink
Arduino/Genuino Uno on COM6



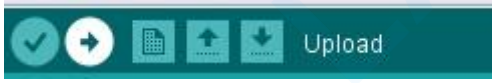
Note: The Board Type and Serial Port here are not necessarily the same as shown in picture. If you are using 2560, then you will have to choose Mega 2560 as the Board Type, other choices can be made in the same manner. And the

Serial Port displayed for everyone is different, despite COM 6 chosen here, it could be COM3 or COM4 on your computer. A right COM port is supposed to be COMX (arduino XXX), which is by the certification criteria.

The Arduino IDE will show you the current settings for board at the bottom of the window.

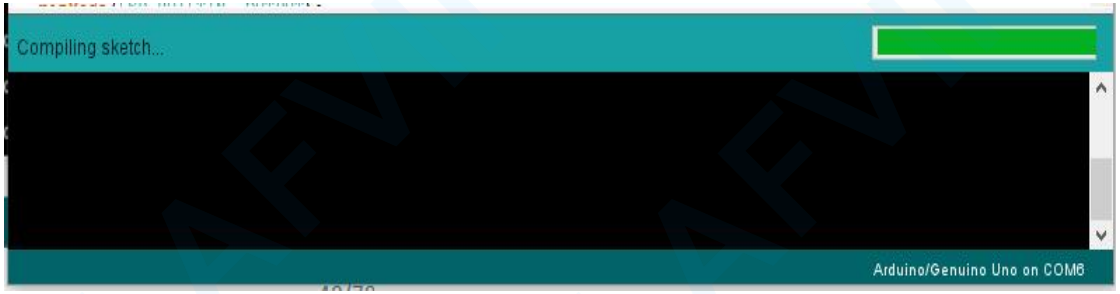


Click on the 'Upload' button. The second button from the left on the toolbar.

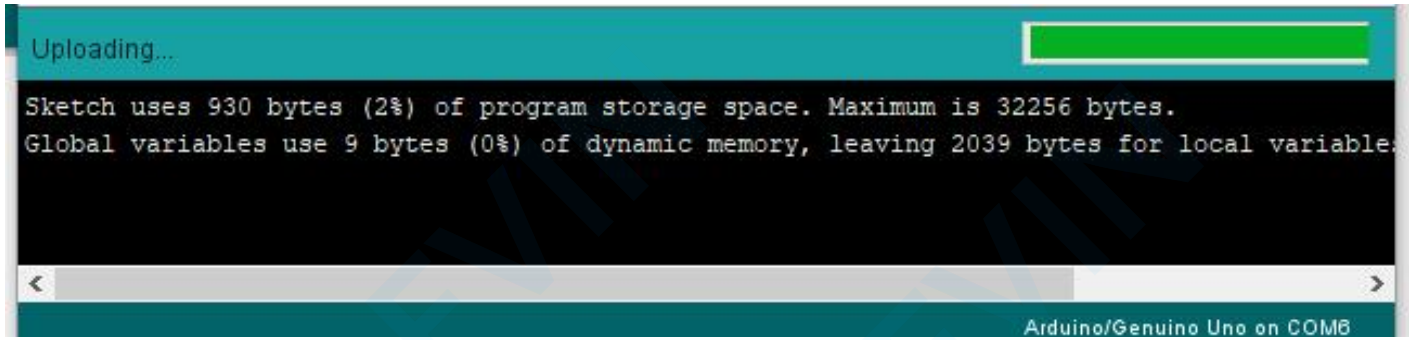


If you watch the status area of the IDE, you will see a progress bar and a series of messages. At first, it will say 'Compiling Sketch...'. This converts the sketch into a format suitable for uploading to the board.

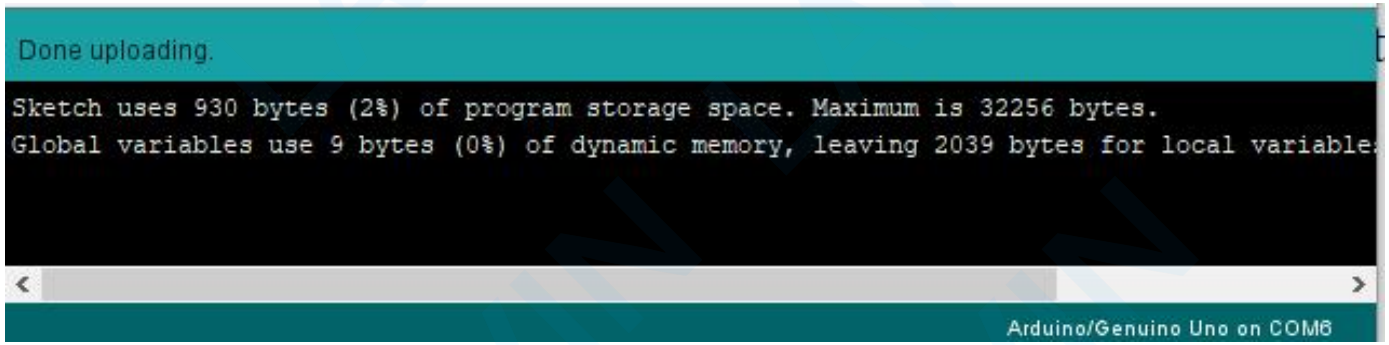
]



Next, the status will change to 'Uploading'. At this point, the LEDs on the Arduino should start to flicker as the sketch is transferred.



Finally, the staus will change to 'Done'.



The other message tells us that the sketch is using 928 bytes of the 32,256 bytes available.After the 'Compiling Sketch..' stage you could get the following error message:

It can mean that your board is not connected at all, or the drivers have not been installed (if necessary) or that the wrong serial port is selected.

If you encounter this, go back to Project 0 and check your installation.

Once the upload has completed, the board should restart and start blinking. Open the code

Note that a huge part of this sketch is composed of comments. These are not actual program instructions; rather, they just explain how the program works. They are there for your benefit.

Everything between /* and */ at the top of the sketch is a block comment; it explains what the sketch is for.
Single line comments start with // and everything up until the end of that line is considered a comment.

The first line of code is: int led = 13;

As the comment above it explains, this is giving a name to the pin that the LED is attached to. This is 13 on most Arduinos,

including the UNO and Leonardo.

Next, we have the 'setup' function. Again, as the comment says, this is executed when the reset button is pressed. It is also executed whenever the board resets for any reason, such as power first being applied to it, or after a sketch has been uploaded.

```
void setup() {
// initialize the digital pin as an output. pinMode(led, OUTPUT);
}
```

Every Arduino sketch must have a 'setup' function, and the place where you might want to add instructions of your own is between the { and the }.

In this case, there is just one command there, which, as the comment states tells the Arduino board that we are going to use the LED pin as an output.

It is also mandatory for a sketch to have a 'loop' function. Unlike the 'setup' function that only runs once, after a reset, the 'loop' function will, after it has finished running its commands, immediately start again.

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

Inside the loop function, the commands first of all turn the LED pin on (HIGH), then 'delay' for 1000 milliseconds (1 second), then turn the LED pin off and pause for another second.

You are now going to make your LED blink faster. As you might have guessed, the key to this lies in changing the parameter in () for the 'delay' command.

// turn the LED off (LOW is the voltage level) // wait for a second
// turn the LED on (HIGH is the voltage level) // wait for a second

```
30 // the loop function runs over and over again forever
31 void loop() {
32   digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the volt
33   delay(500)                         // wait for a second
34   digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the vo
35   delay(500)                         // wait for a second
36 }
```

This delay period is in milliseconds, so if you want the LED to blink twice as fast, change the value from 1000 to 500. This would then pause for half a second each delay rather than a whole second.

Upload the sketch again and you should see the LED start to blink more quickly

3.Getting Started with Mixly

3.1 Introduction of Mixly Software

Mixly is a free open-source graphical Arduino programming software, based on Google's Blockly graphical programming framework, and developed by Mixly Team@ BNU. It is a free open-source graphical programming tool for creative electronic development; a complete support ecosystem for creative e-education; a stage for maker educators to realize their dreams.Although there is an Ardublock graphical programming software launched by Arduino official, Ardublock is not perfect enough, and many common functions cannot be realized.

Design Concept:

(1) Usability

Mixly is designed to be completely green. Currently. It can run on Windows xp and above.

(2) Simplicity

Mixly uses the Blockly graphical programming engine to replace complex text manipulation with graphical building blocks, providing a good foundation for beginners to get started quickly.

1 Use the different color icons to represent different types of functional blocks, very convenient for users to classify.

2 Provide default options in the composite function block to effectively reduce the number of user drags.

3 Integrate all the features of the software in the same interface.

4 Provide the reference tutorial and code examples.

(3) Functionality

It has versatile functions. Mixly can almost implement all the functions that Arduino IDE has. Support all official development boards of arduino.

(4) Continuity

The goal of the graphical programming system is definitely not to replace the original text programming method,but to better understand the programming principles and program thinking through graphical programming, and lay the foundation for future text programming. It is also the design philosophy for Mixly. More continuous content has been added to the design of the software to protect the user's learning outcomes. To be specific, it includes the introduction of variable types, the consistency of text programming as much as possible in the design of the module, and the support of

both graphical and text programming.

(5) Ecological

The most important design concept of Mixly is its ecological feature, which can distinguish it from other Arduino graphical programming.In order to achieve sustainable development, Mixly is designed to allow manufacturers to develop their own unique modules.But users require JavaScript programming foundation to make this part of the module).It also allows users directly use Mixly's graphical programming function to generate common modules (such as LED digital display, buzzer broadcast, etc. Users are able to make this part of the module only using Mixly). Both of the two kinds of modules mentioned above can be imported into the Mixly system through the "Import" function, thereby realizing the user's own value in the popularity of Mixly software.

User Groups:

From the above design concept, it can be seen that Mixly is suitable for primary and secondary school students to cultivate programming thinking. It is also available for quick programming when creating a work.it is good for those lovely friends who don't want to learn text programming, but want to do some small works with intelligent control.

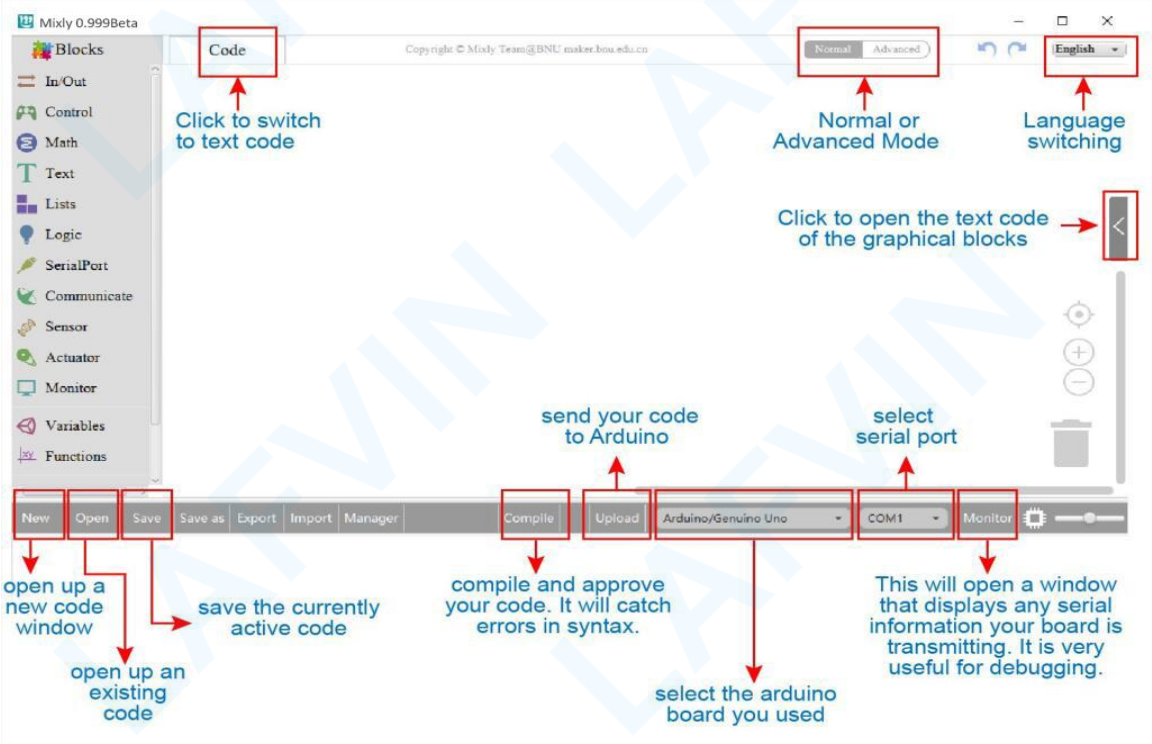
Mixly Blocks Functions:

System Functions

Look at the main interface of Mixly, it includes five parts, that is, Blocks selection, code edit, text code (hidden), system function and message prompt area. Shown below.

Some common functions:

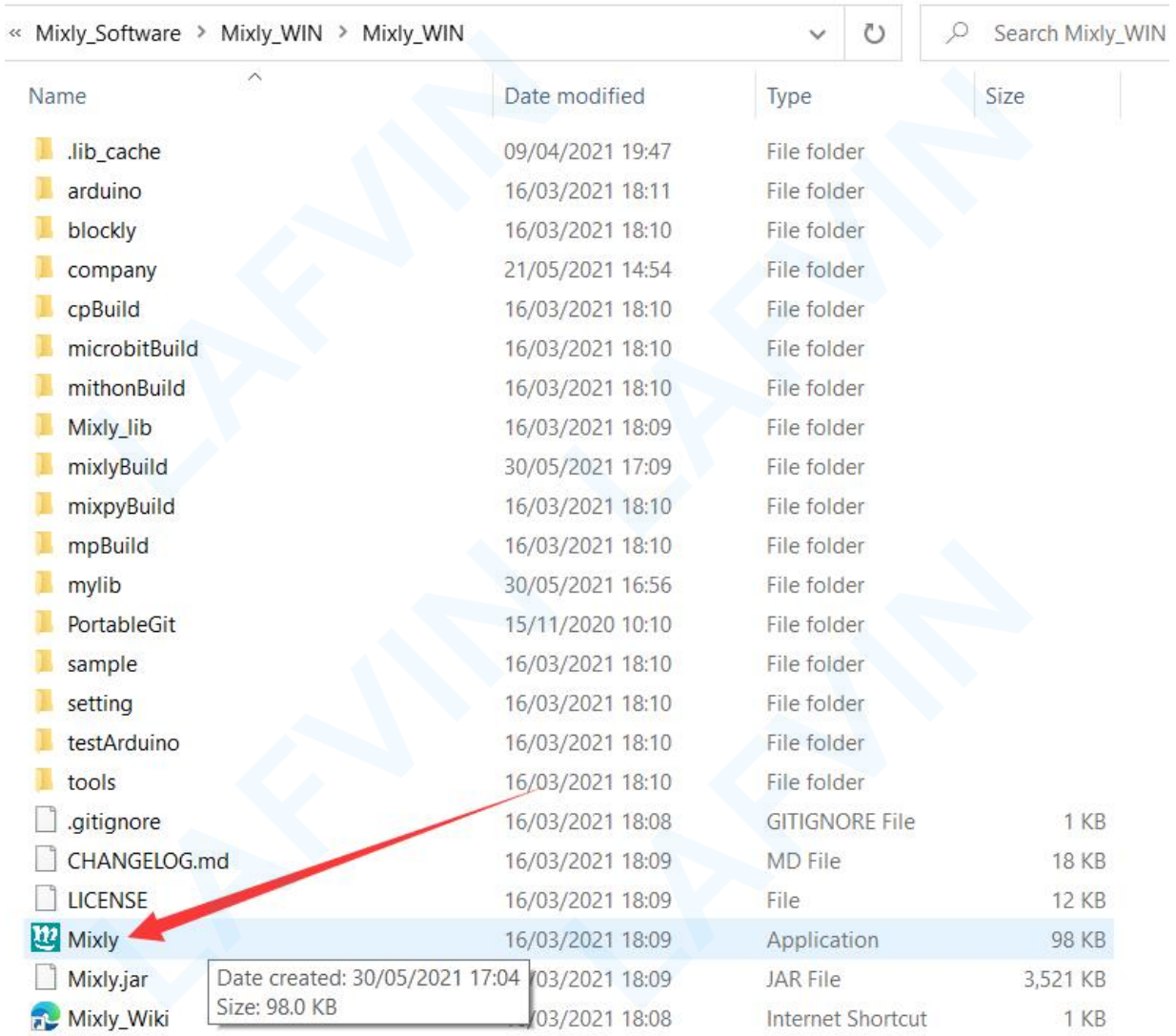
Through this interface, you can complete the code compile、upload、save and manage. It support four remove methods: drag it left out code window, or drag to Recycle Bin, delete key, or right-click to delete block. It supports four languages: English、Español (Spanish)、中文简体(Chinese Simplified)、中文繁体(Chinese Traditional).



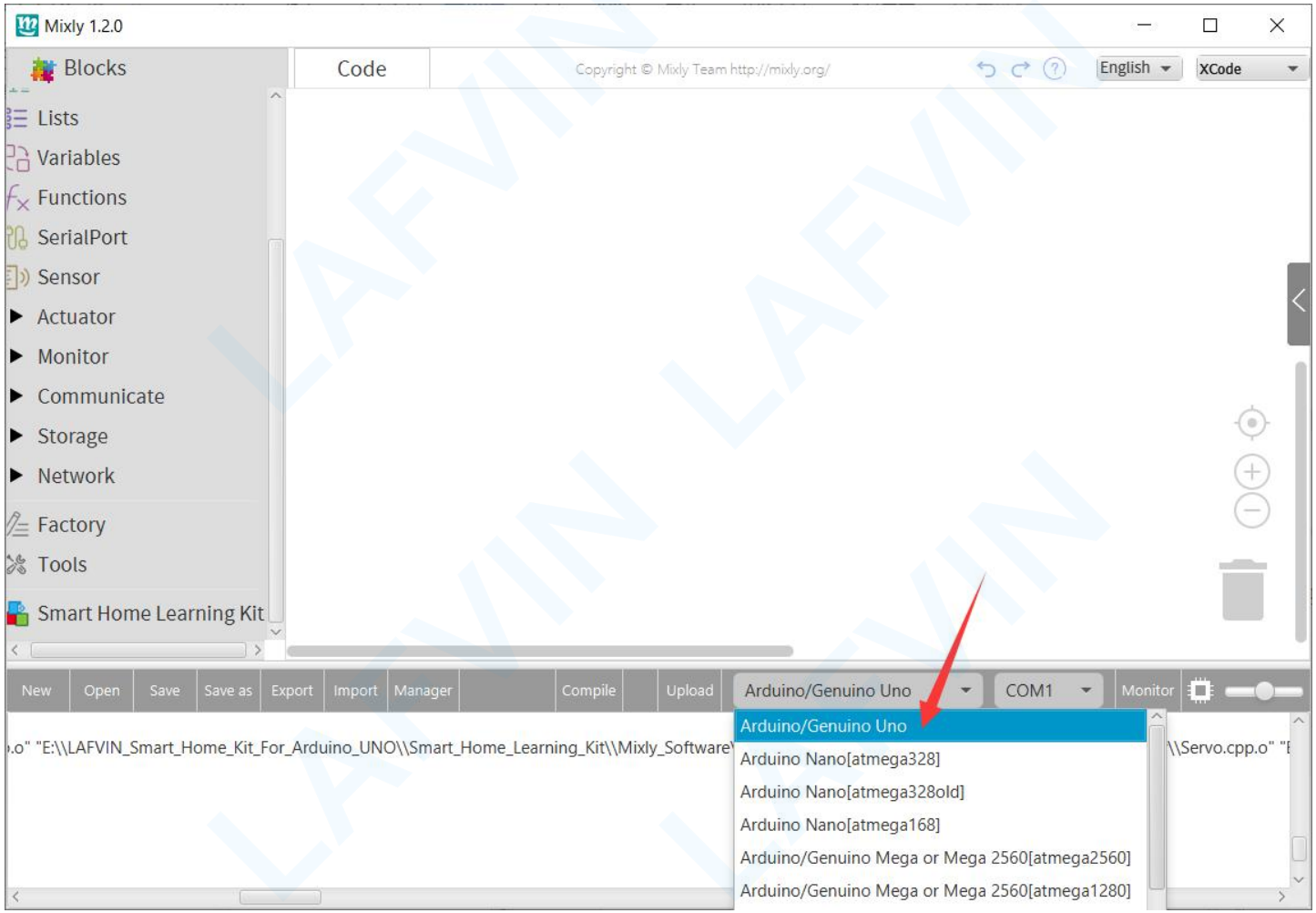
3.2 How to Install Mixly Software

First decompress the mixly programming software file

After unzipping you will get the following files(Note: Do not include special characters such as Chinese, spaces, brackets, etc. in the unzipped directory.The name of the uncompressed path cannot have a space bar. If the named name needs a separator, you can use an underscore to replace the space bar, for example,you should name the folder Mixly_Sofewave instead of Mixly Sofewave)

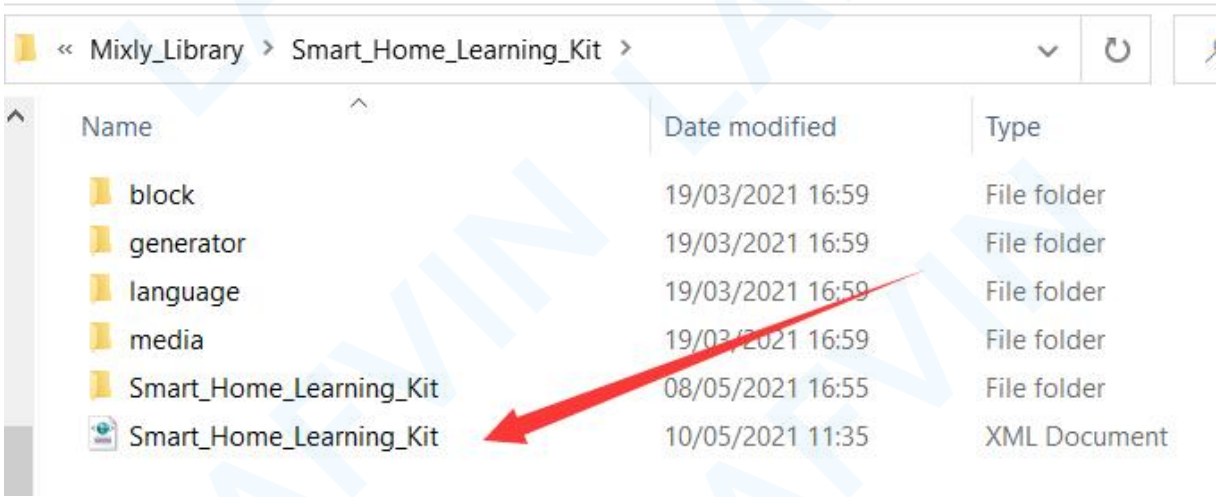

« Mixly_Software › Mixly_WIN › Mixly_WIN
Search Mixly_WIN
Name
Date modified
Type
Size
.lib_cache 09/04/2021 19:47 File folder
arduino 16/03/2021 18:11 File folder
blockly 16/03/2021 18:10 File folder
company 21/05/2021 14:54 File folder
cpBuild 16/03/2021 18:10 File folder
microbitBuild 16/03/2021 18:10 File folder
mithonBuild 16/03/2021 18:10 File folder
Mixly_lib 16/03/2021 18:09 File folder
mixlyBuild 30/05/2021 17:09 File folder
mixpyBuild 16/03/2021 18:10 File folder
mpBuild 16/03/2021 18:10 File folder
mylib 30/05/2021 16:56 File folder
PortableGit 15/11/2020 10:10 File folder
sample 16/03/2021 18:10 File folder
setting 16/03/2021 18:10 File folder
testArduino 16/03/2021 18:10 File folder
tools 16/03/2021 18:10 File folder
.gitignore 16/03/2021 18:08 GITIGNORE File 1 KB
CHANGELOG.md 16/03/2021 18:09 MD File 18 KB
LICENSE 16/03/2021 18:09 File 12 KB
Mixly 16/03/2021 18:09 Application 98 KB
Mixly.jar 03/2021 18:09 JAR File 3,521 KB
Mixly_Wiki 03/2021 18:08 Internet Shortcut 1 KB
Date created: 30/05/2021 17:04
Size: 98.0 KB

Double-click the mixly application file to open the software. And select the main control board as **Arduino/Genuino Uno**



3.3 How to Add Mixly Libraries

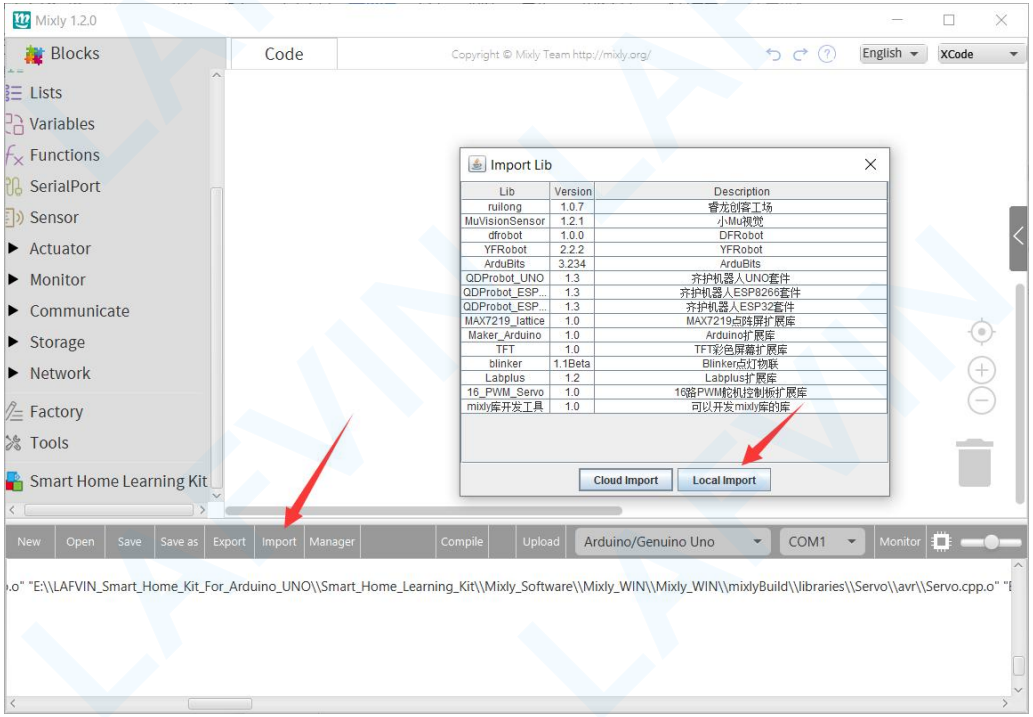
The mixly library file is an integrated graphical code customized for a certain control board. For example, we have written a mixly library file for smart home learning kit. You can directly load it into mixly software and use it directly, which brings convenience to your programming process.The general library file contains the following files, the last .xml file is the readable library file of the mixly software

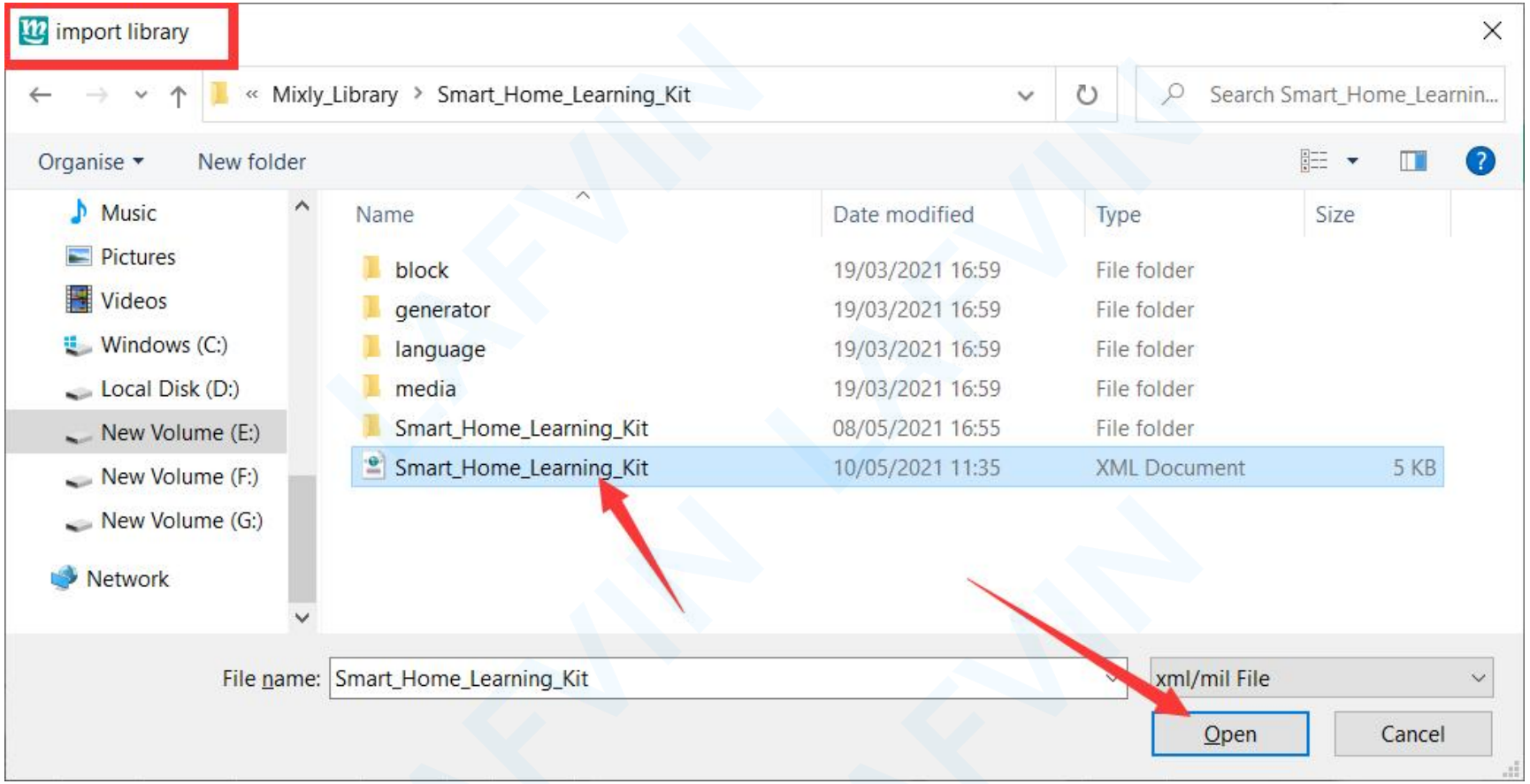


(Note:After completing the installation of the mixly software, the library file <Smart Home Learning Kit> already contains the automatic addition to the programming software, you can use it directly without adding and adding repeatedly.)

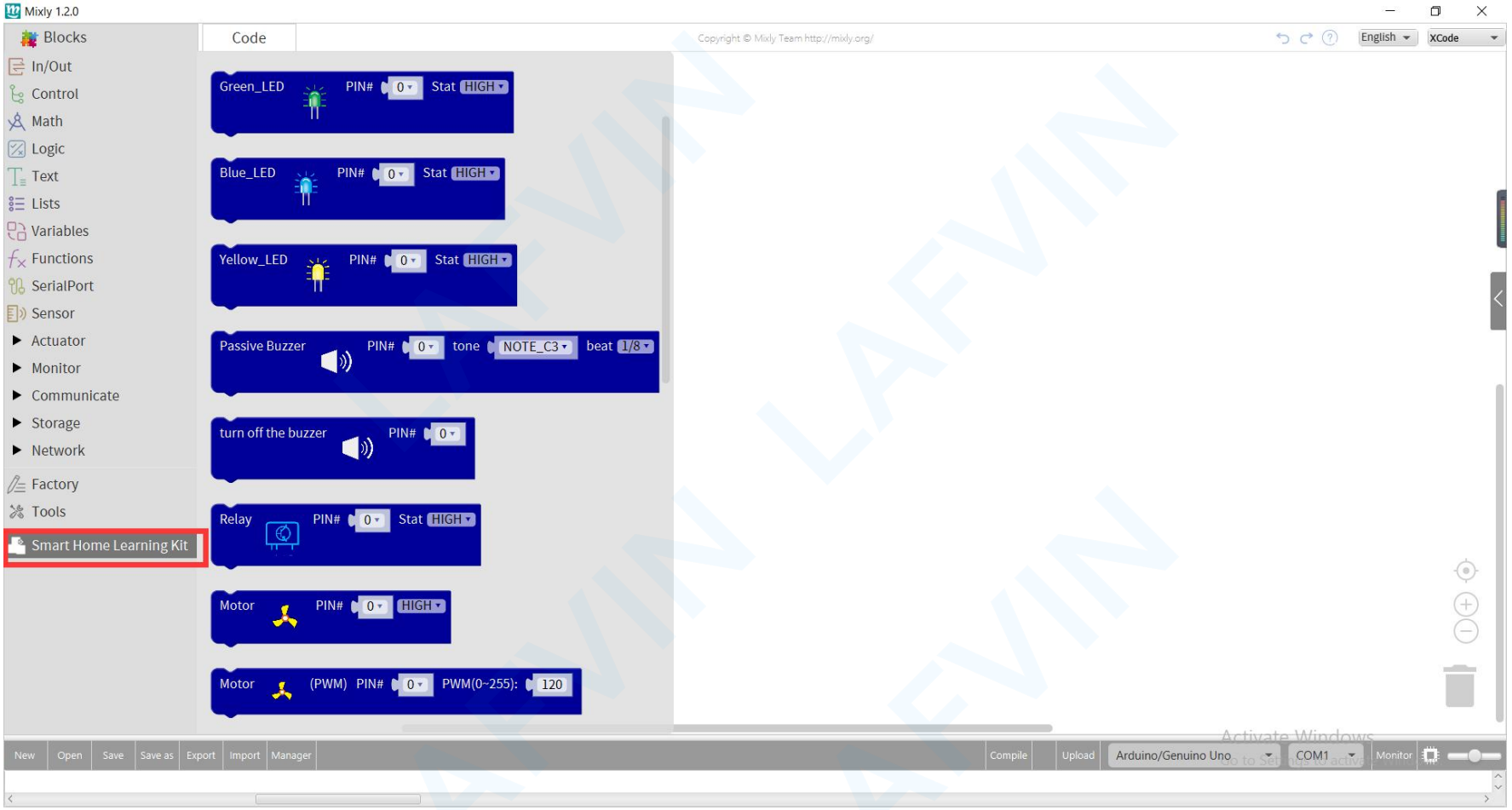
If you need to accidentally delete the library file, or you want to add other functional library files, you can refer to the following steps to add library files.

After opening the mixly software, select "**import**" and click **local load** .then select the path where the "**Smart Home Learning Kit.xml**" folder is located,select the file --"**mart Home Learning Kit.xml**", and click "open"



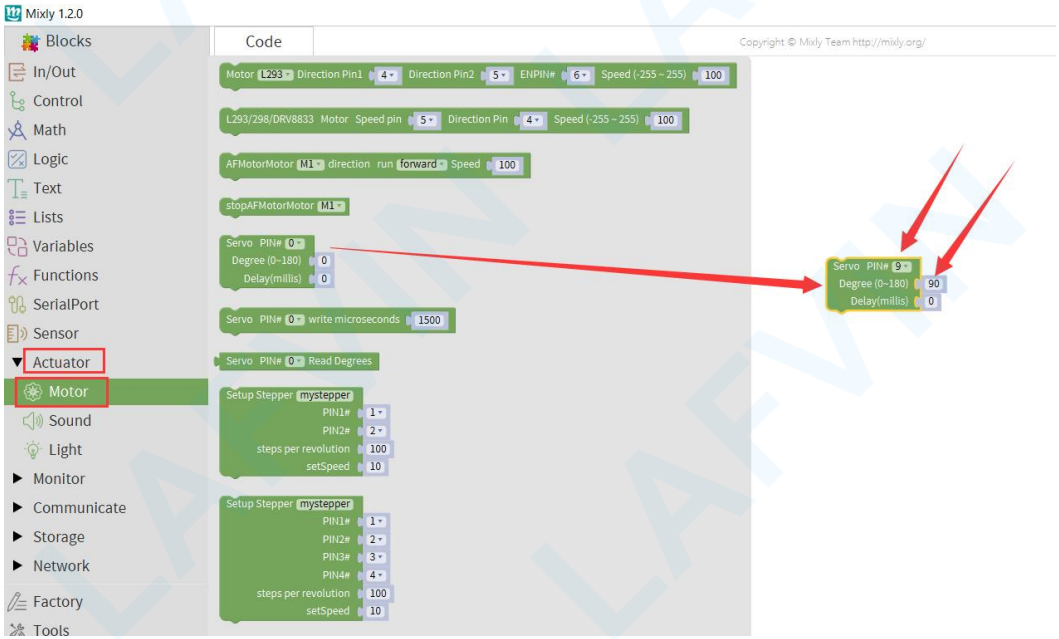


The figure below shows that the ”Smart_Robot_Tank”library file has been successfully imported into the mixly software.

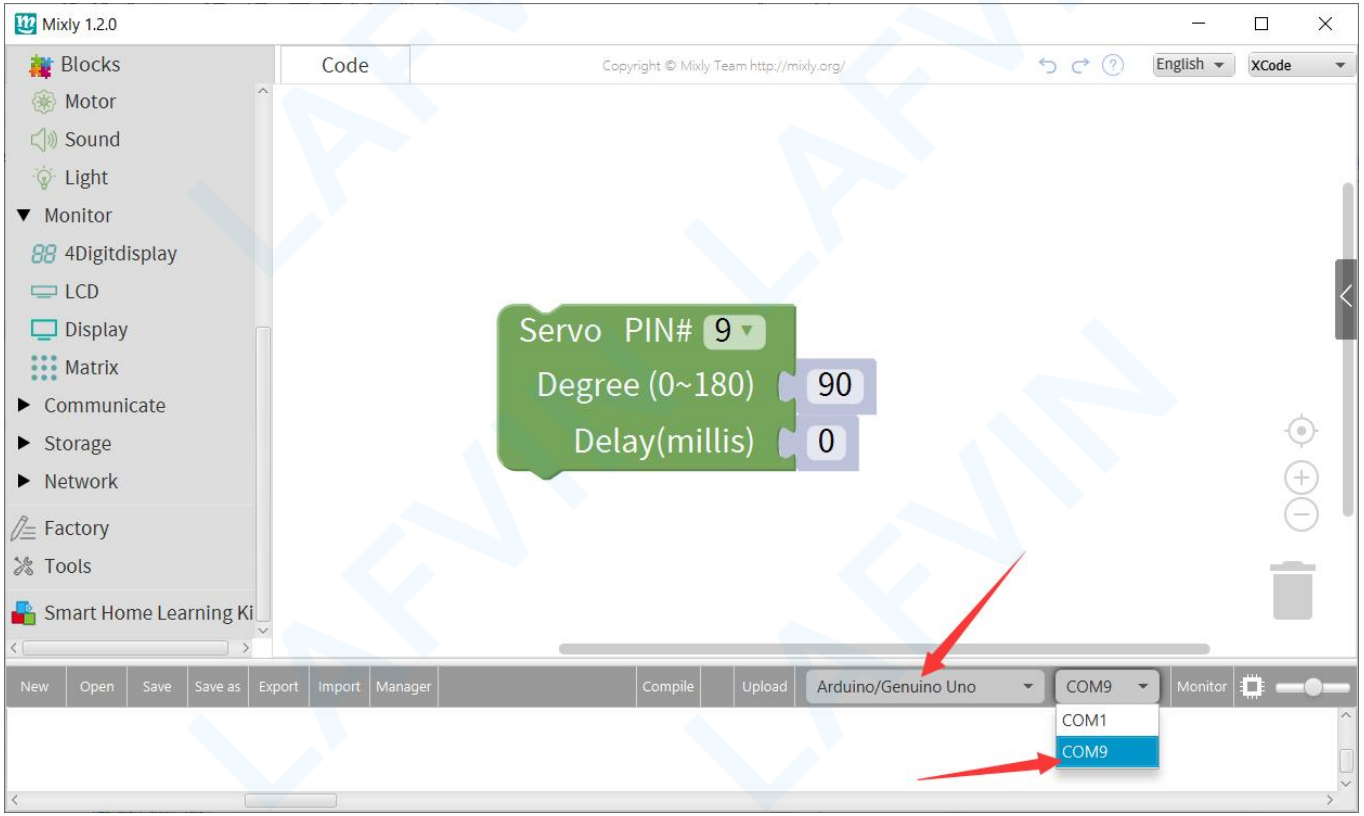

Mixly 1.2.0
Blocks
In/Out
Control
Math
Logic
Text
Lists
Variables
Functions
SerialPort
Sensor
Actuator
Monitor
Communicate
Storage
Network
Factory
Tools
Smart Home Learning Kit
Code
Green_LED PIN# 0 Stat HIGH
Blue_LED PIN# 0 Stat HIGH
Yellow_LED PIN# 0 Stat HIGH
Passive Buzzer PIN# 0 tone NOTE_C3 beat 1/8
turn off the buzzer PIN# 0
Relay PIN# 0 Stat HIGH
Motor PIN# 0 HIGH
Motor (PWM) PIN# 0 PWM(0~255): 120
New
Open
Save
Save as
Export
Import
Manager
Compile
Upload
Arduino/Genuino Uno
COM1
Monitor

Next we write a program on the software and upload it to main controller of smart home kit.Try to write a code to control the rotation of the servo motor to 90 degrees.(and at the same time you need to connect the servo to the digital interface D9 of the sensor shield)

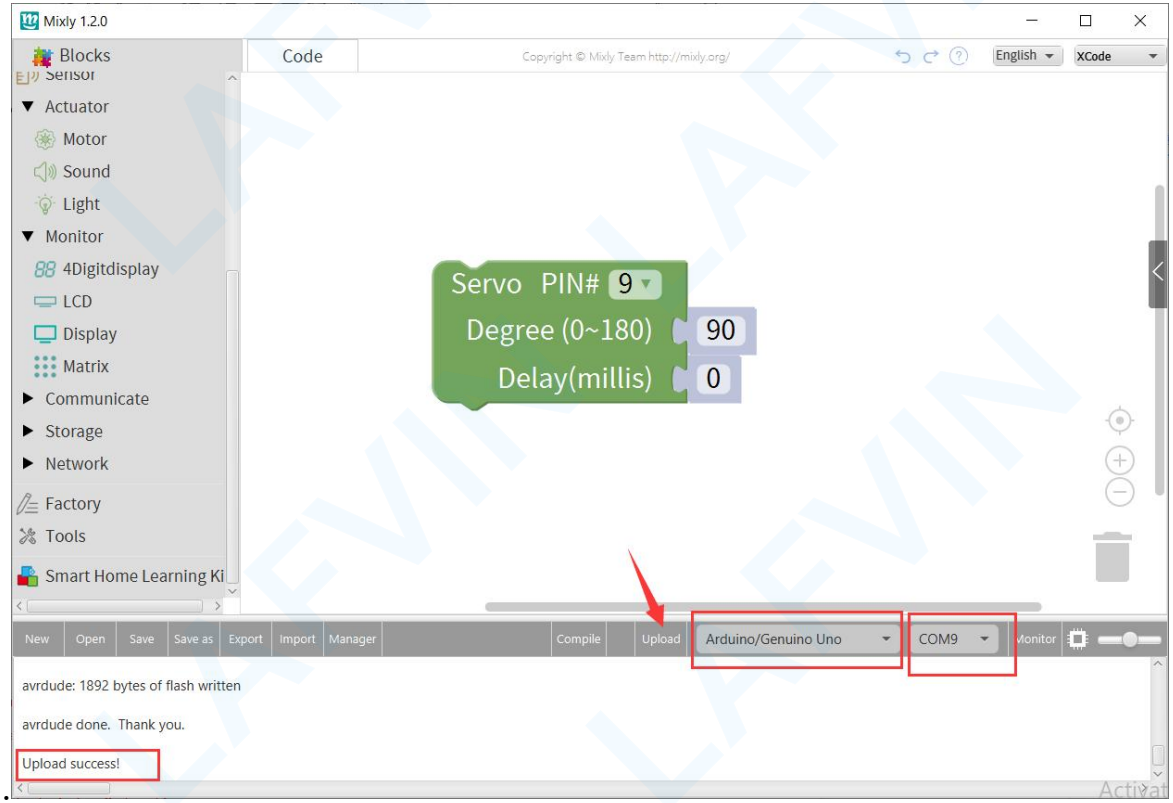
Take out the servo program block from the subject bar of the actuator to the program area, and modify the PIN variable to 9.modify the degree variable to 90.



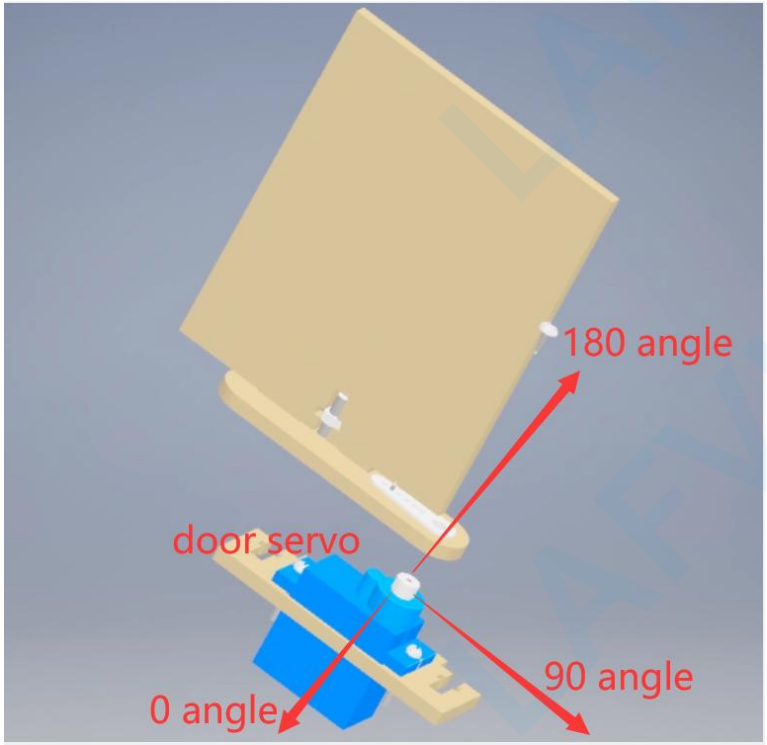
Connect the Arduino UNO development board to the computer with a USB data cable, and turn on the power switch. Select the development board type as "Arduino UNO" on the software and you will see a new connection serial port "COM9" appears.

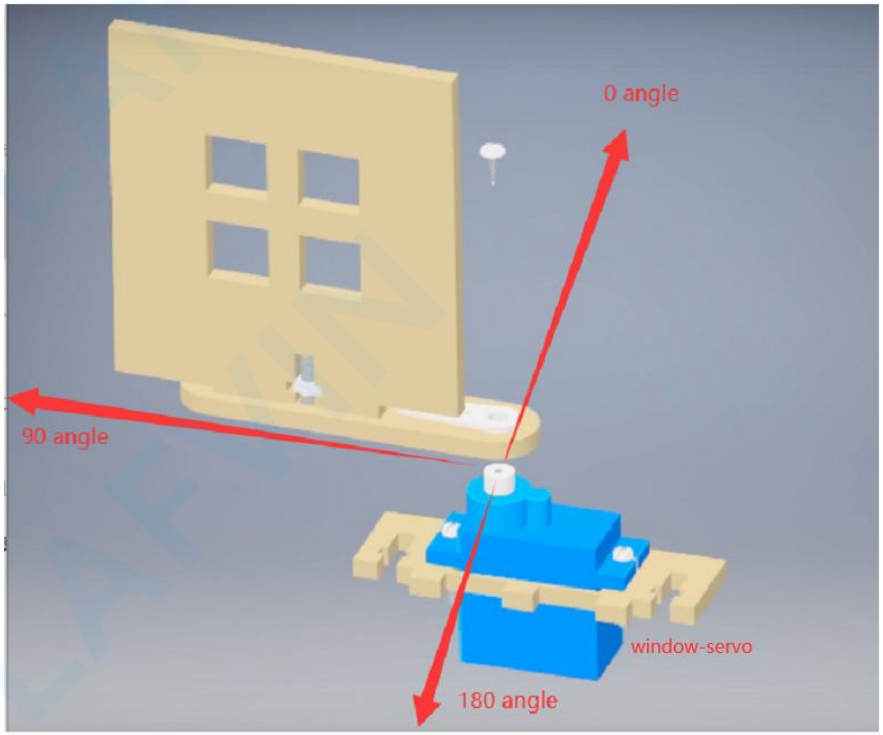


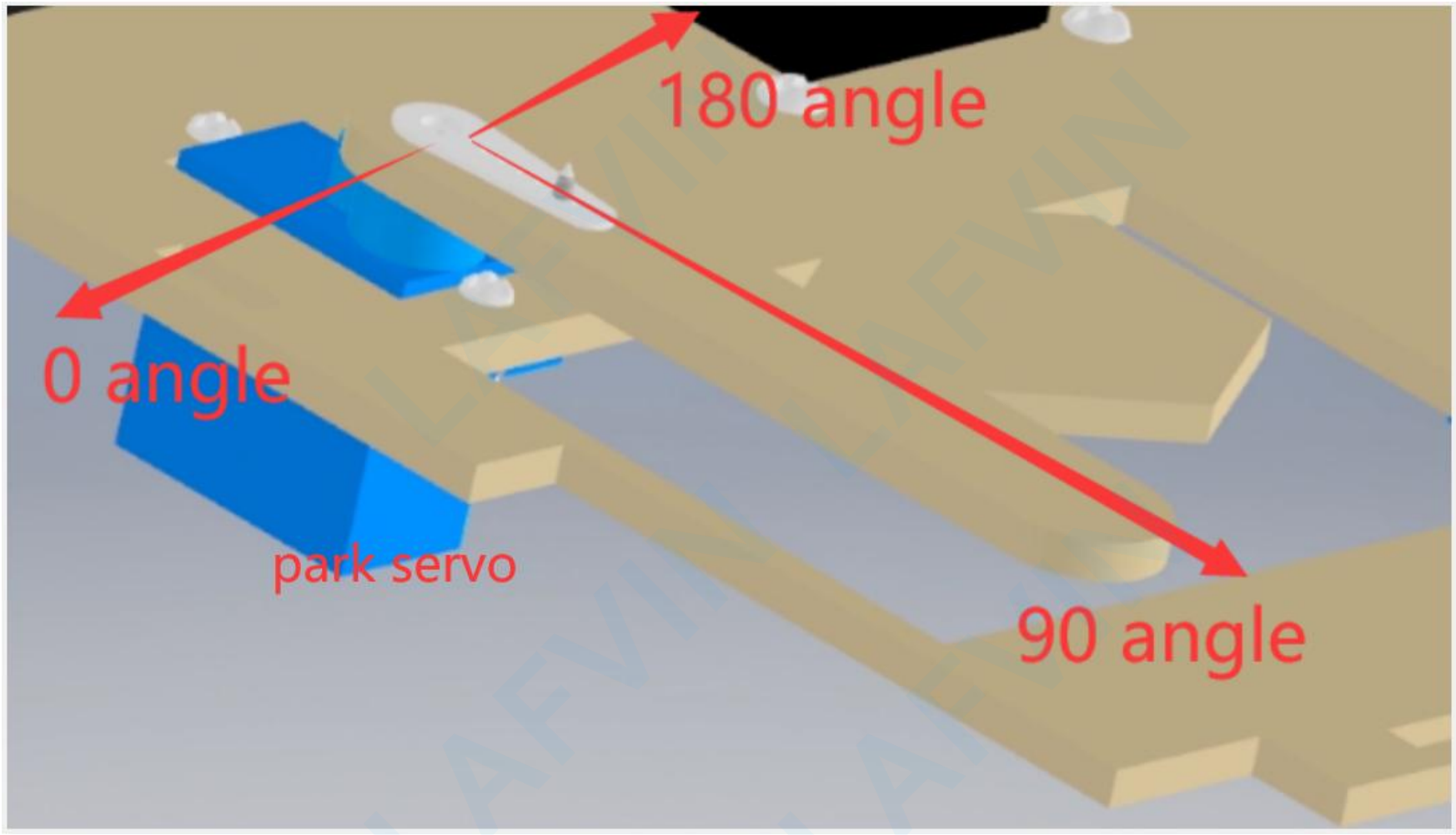
After completing the program upload, connect the servo motor to the digital interface D9, and the servo motor will execute the command to rotate to the 90 degree position. (This is also the default position of the servo motor when it leaves the factory)



There are three servo motors that need to be installed in the learning kit. The installation angle of the servo motor in the installation video is the default 90 degrees. If you accidentally damage this angle, you can restore the default angle of the servo motor by uploading the program just now (90 degrees), and then reinstall strictly according to the direction, angle and position in the installation video.

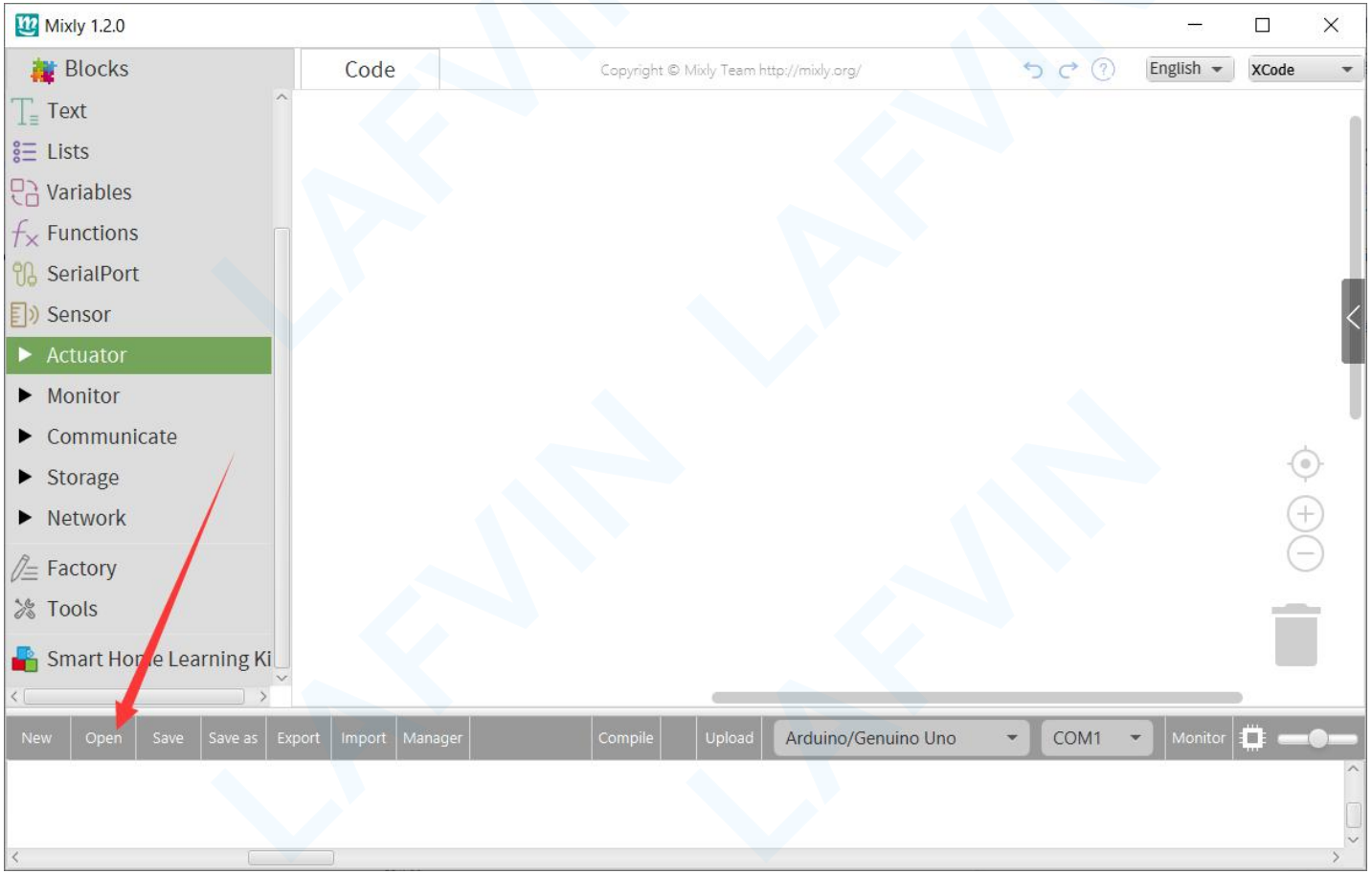


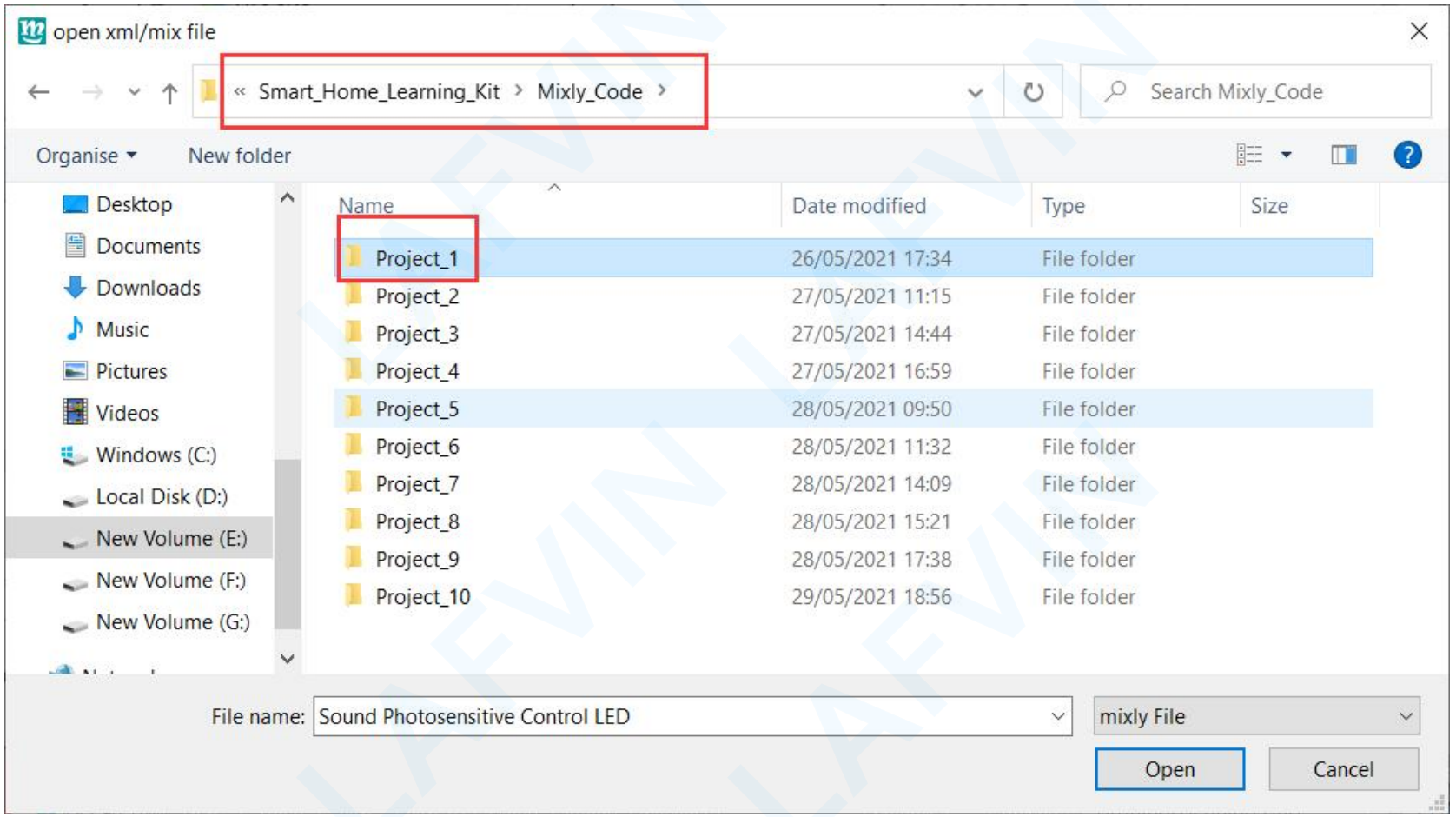


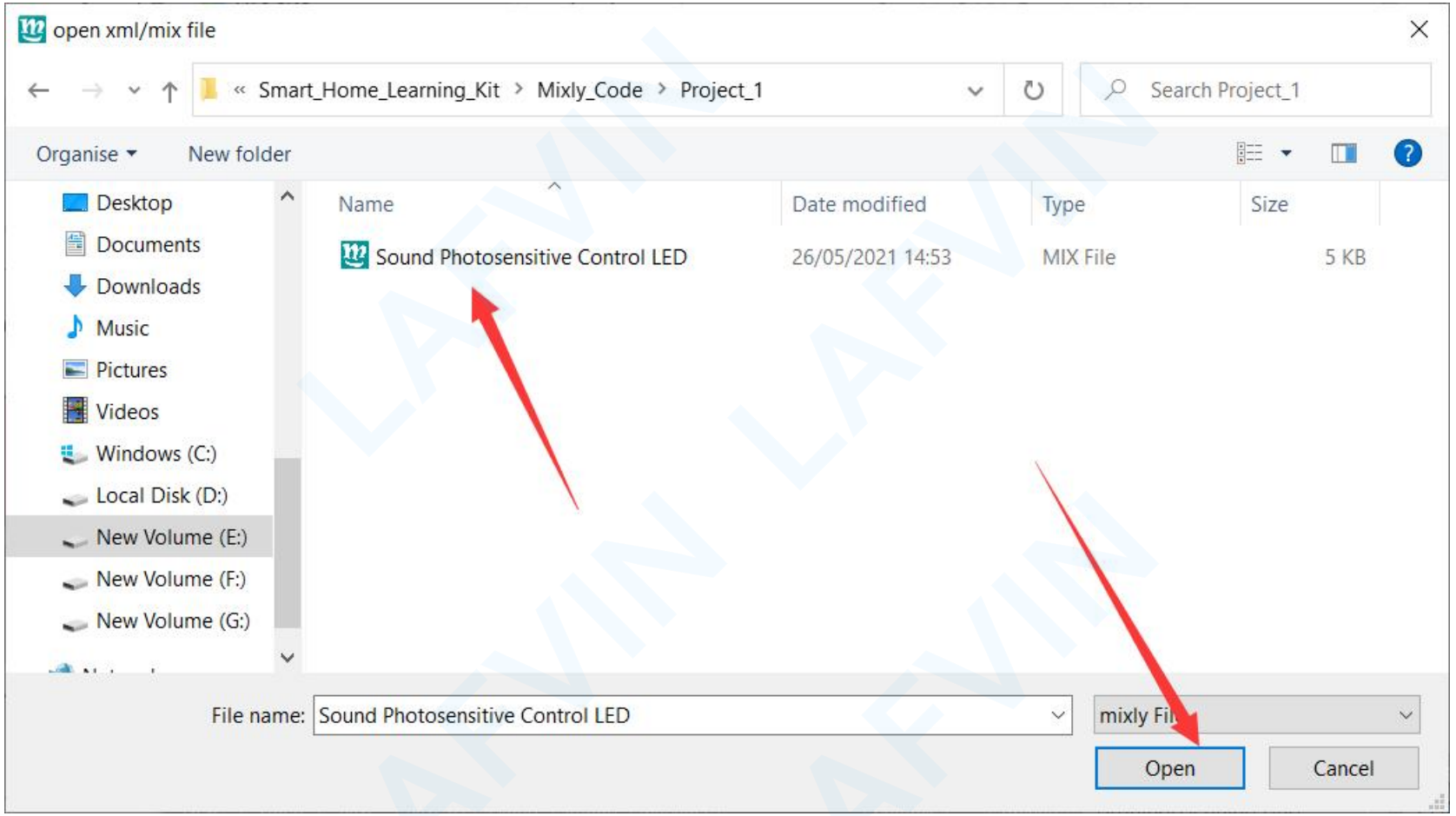

180 angle
0 angle
park servo
90 angle

3.4 How to open the reference program

Click "OPEN" on the mixly software and select the reference program.

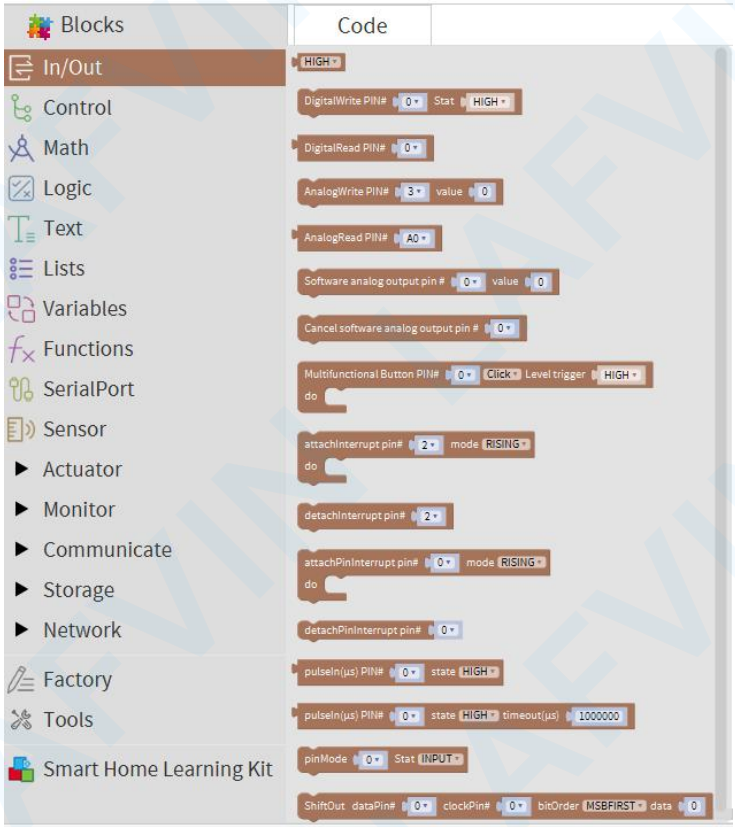



open xml/mix file
Smart_Home_Learning_Kit
Mixly_Code
Search Mixly_Code
Organise
New folder
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
Local Disk (D:)
New Volume (E:)
New Volume (F:)
New Volume (G:)
Name
Date modified
Type
Size
Project_1 26/05/2021 17:34 File folder
Project_2 27/05/2021 11:15 File folder
Project_3 27/05/2021 14:44 File folder
Project_4 27/05/2021 16:59 File folder
Project_5 28/05/2021 09:50 File folder
Project_6 28/05/2021 11:32 File folder
Project_7 28/05/2021 14:09 File folder
Project_8 28/05/2021 15:21 File folder
Project_9 28/05/2021 17:38 File folder
Project_10 29/05/2021 18:56 File folder
File name:
Sound Photosensitive Control LED
mixly File
Open
Cancel


open xml/mix file
Smart_Home_Learning_Kit
Mixly_Code
Project_1
Search Project_1
Organise
New folder
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
Local Disk (D:)
New Volume (E:)
New Volume (F:)
New Volume (G:)
Name
Date modified
Type
Size
Sound Photosensitive Control LED
26/05/2021 14:53
MIX File
5 KB
File name:
Sound Photosensitive Control LED
Open
Cancel

The use of each statement block is described below

In/Out Block:



NO.	BLOCK ICON	DEFINITION
1		Returns HIGH or LOW voltage
2		Write digital value to a specific Port. Digital Output: set the HIGH or LOW output for IO pins
3		Returns a digital value of a specific Port. Digital IO Read Pin, generally used to read the HIGH or LOW level detected by Digital sensor

4

Write analog value between 2 and 255 to a specific Port. Analog Output: set the Analog value output by Analog IO pins (0~255).

5

Returns value between 0 and 1023 of a specific Port. Analog IO Read Pin, generally used to read the Analog value detected by Analog sensor.

6

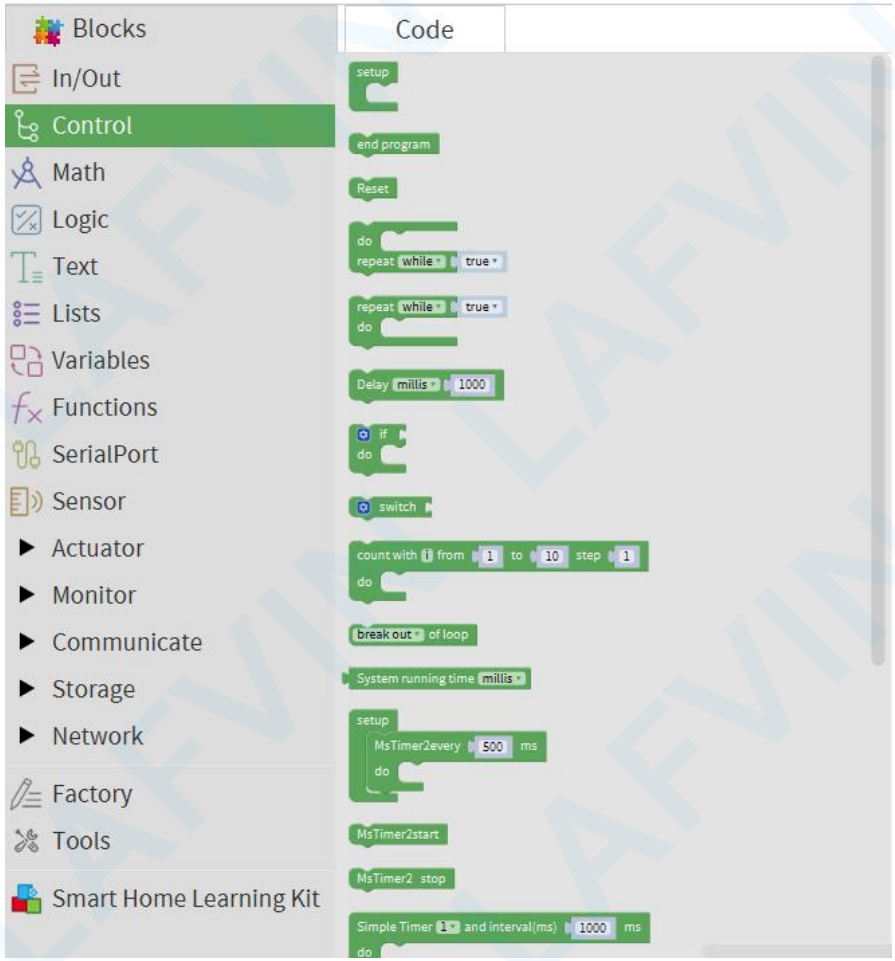
External Interrupts function, with three trigger interrupt modes RISING, FALLING, CHANGE.

7

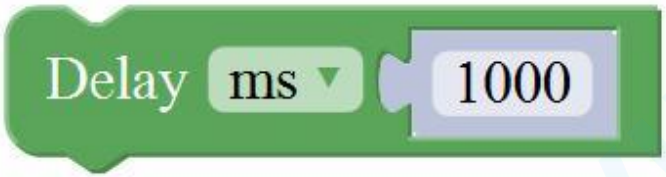
Detachs interrupt to a specific Port. Turn off the given interrupt function.

8		Set the IO pins as Output or Input state
9		Read the continuous time of HIGH or LOW pulse from IO pins (generally used for ultrasonic ranging)

Control Block:



NO.	BLOCK ICON	DEFINITION
1		Initialization (run only once)
2		End the program, means the program will stop running when use this block.
3		Delay function, click to select **ms** or **us** (pause the program for the amount of time (in milliseconds)



specified as parameter. There are 1000 milliseconds in a second.)

4

if_do function (first evaluate a value be true or false, if a value is true, then do some statement. You can click the blue gear icon to select the **else if** block or **else** block.)

5

switch function. You can click the blue gear icon to select the **case** block or **default** block. (used to evaluate several programs then execute the corresponding function matched with program.)

6

Equal to **for** statement.

7 

A **while** loop statement.

8 

break function, used to exit from the containing loop.

9

millis() function, returns the system running time since the program started. (The unit can be **ms** (milliseconds) or **μs**（ m icrosecond)).

10

Timer interrupt function, that is, set a trigger interrupt for the amount of time (in milliseconds) specified as parameter.

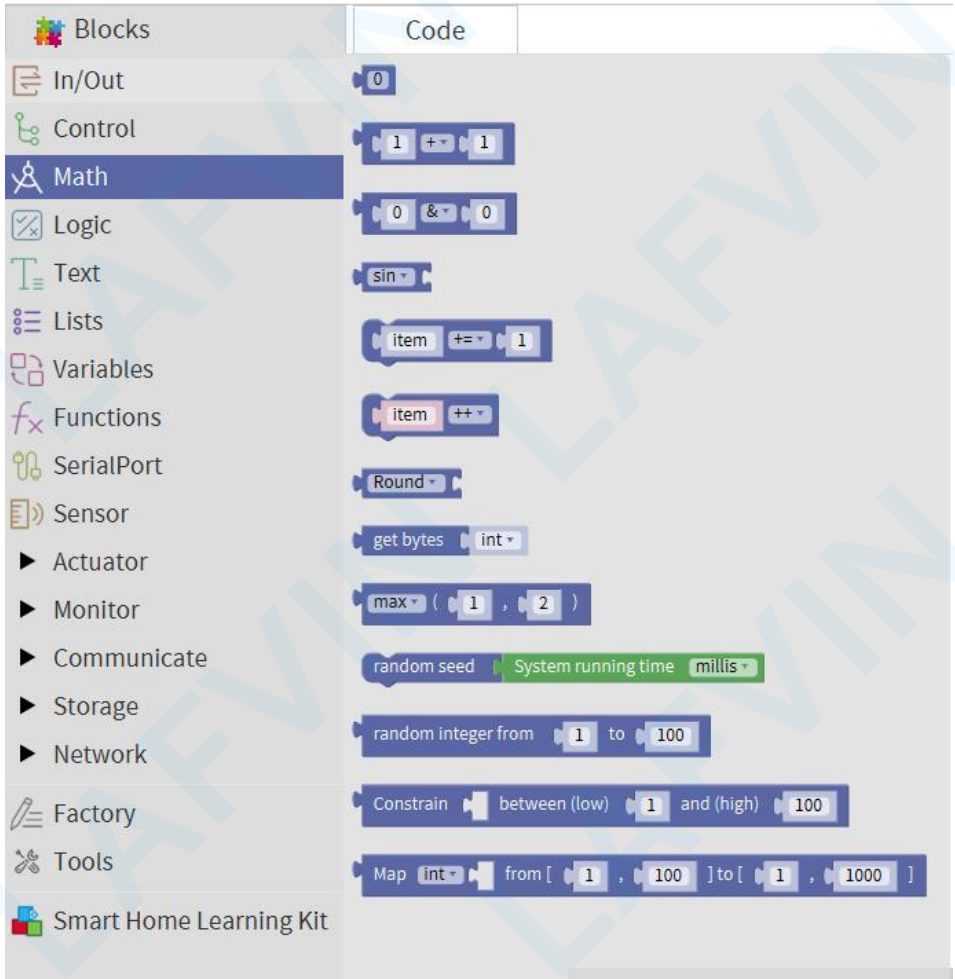
11 

Timer interrupt start block

12 

Timer interrupt stop block

Math Block:


Blocks
In/Out
Control
Math
Logic
Text
Lists
Variables
Functions
SerialPort
Sensor
Actuator
Monitor
Communicate
Storage
Network
Factory
Tools
Smart Home Learning Kit
Code
0
1 + 1
0 & 0
sin
item += 1
item ++
Round
get bytes int
max (1 , 2)
random seed System running time millis
random integer from 1 to 100
Constrain between (low) 1 and (high) 100
Map int from [1 , 100] to [1 , 1000]

NO.	BLOCK ICON	DEFINITION
1		A number
2		Click to select the Arithmetic Operators: + (addition) ; − (subtraction) ; x (Multiplication); ÷ (division); % (remainder); ^ (bitwise xor)
3		Click to select the & (bitwise end); I (bitwise or); << (bitshift left); >> (bitshift right)

4

Click to select the **sin; cos; tan; asin; acos; atan; ln; log10; e^; 10^; ++ (increment) ; --(decrement)**

5

Click to select the **Round; Ceil; Floor; abs; sq; sqrt**

Round: Returns the integer part a number using around. **Ceil:** Returns the integer part a number using ceil. **Floor:** Returns the integer part a number using floor. **abs:** Return the absolute value of a number. **sq:** Return the square of a number. **sqrt:** Return the

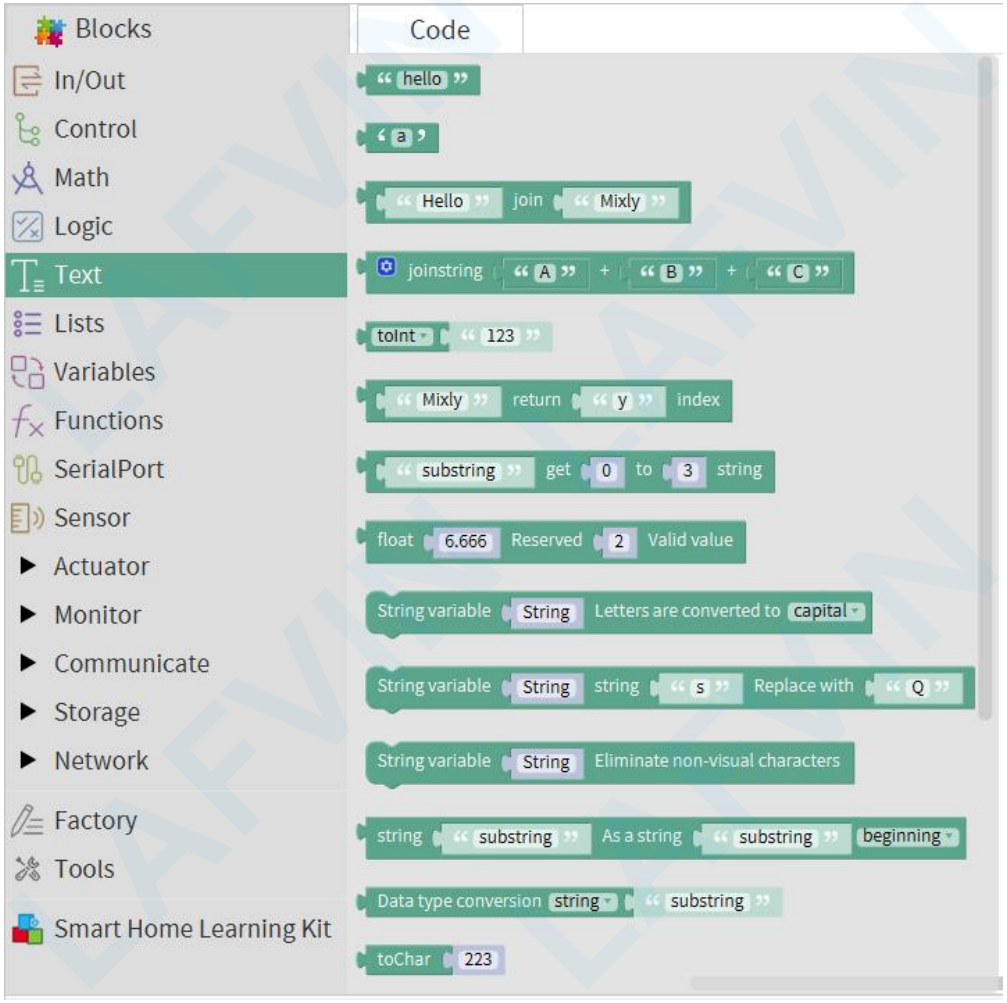
square root of a number.

6

If select the max, returns the larger number;

if select the min, returns the smaller number.

Text Block:


Blocks
In/Out
Control
Math
Logic
Text
Lists
Variables
Functions
SerialPort
Sensor
Actuator
Monitor
Communicate
Storage
Network
Factory
Tools
Smart Home Learning Kit
Code
hello
a
Hello join Mixly
joinstring A + B + C
toInt 123
Mixly return y index
substring get 0 to 3 string
float 6.666 Reserved 2 Valid value
String variable String Letters are converted to capital
String variable String string s Replace with Q
String variable String Eliminate non-visual characters
string substring As a string substring beginning
Data type conversion string substring
toChar 223

NO.	BLOCK ICON	DEFINITION
1		character string: a letter, word, or line of text.
2		A character
3		Creates a piece of text by joining together two piece of text. (Here Hello join Mixly equals HelloMixly)
4		Converts a string into an integer or an float.

5		Returns the char corresponding to an ASCII code (Decimal number 97 corresponding to a)
6		Returns the ASCII code corresponding to a char.
7		Converts a number into a string.
8		Calculates the length of a string
9		Output the char of a string (the char at 0 of hello is h)

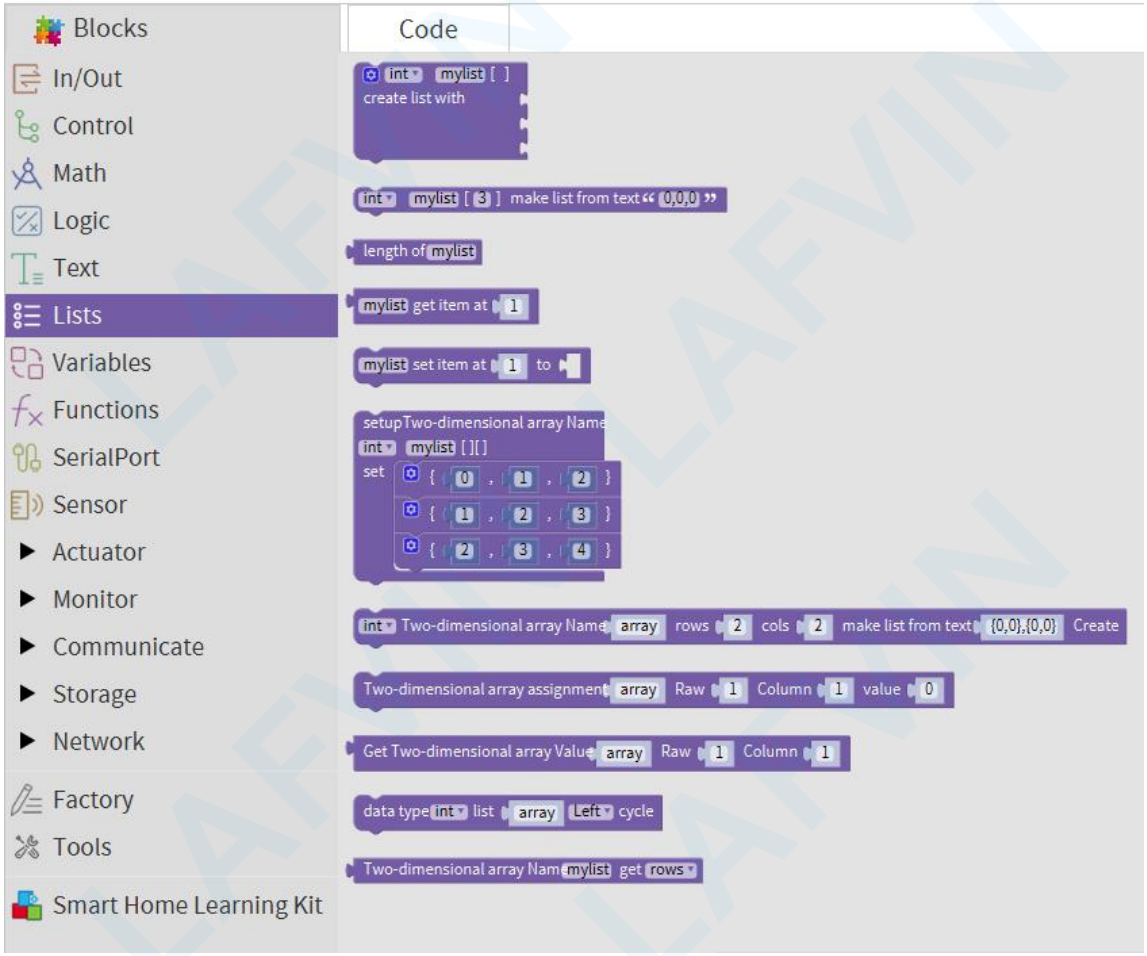
10

The first string equals or startsWith or endsWith the second string, returns 1, otherwise returns 0. (if equals, both strings are abc, returns 1.)

11

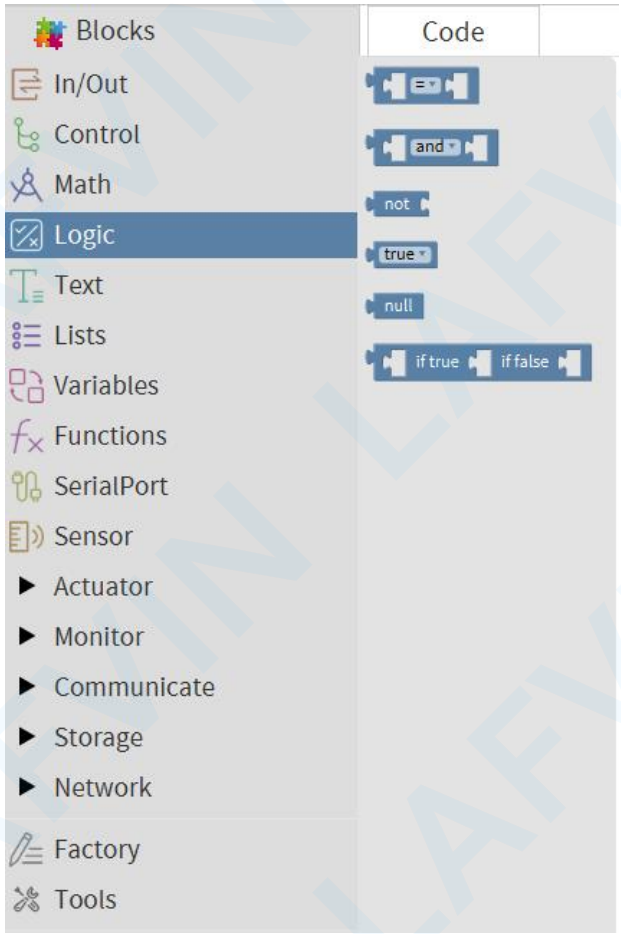
Returns a decimal value of the first string subtracts the second string.

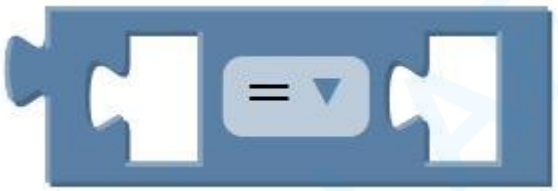
List Block:

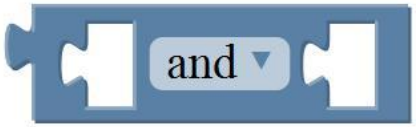

Blocks
In/Out
Control
Math
Logic
Text
Lists
Variables
Functions
SerialPort
Sensor
Actuator
Monitor
Communicate
Storage
Network
Factory
Tools
Smart Home Learning Kit
Code
int mylist [ ]
create list with
int mylist [3] make list from text " 0,0,0 "
length of mylist
mylist get item at 1
mylist set item at 1 to
setup Two-dimensional array Name
int mylist [ ][ ]
set { 0 , 1 , 2 }
{ 1 , 2 , 3 }
{ 2 , 3 , 4 }
int Two-dimensional array Name array rows 2 cols 2 make list from text {0,0},{0,0} Create
Two-dimensional array assignment array Raw 1 Column 1 value 0
Get Two-dimensional array Value array Raw 1 Column 1
data type int list array Left cycle
Two-dimensional array Nam mylist get rows

NO.	BLOCK ICON	DEFINITION
1		Create a list with any number of items
2		Creats a list from a text. (int mylist [ ]={0,0,0};)
3		Returns the length of a list
4		Returns the value of at the specified position in a list.
5		Sets the value of at the specified position in a list. Set the first item in mylist to another item.

Logic Block:


Blocks
In/Out
Control
Math
Logic
Text
Lists
Variables
Functions
SerialPort
Sensor
Actuator
Monitor
Communicate
Storage
Network
Factory
Tools
Code
=
and
not
true
null
if true
if false

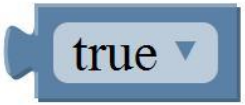
NO.	BLOCK ICON	DEFINITION
1		**logic comparision** =: Return true if both inputs equal each other. ≠: Return true if both inputs are not equal to each other. <: Return true if the first input is smaller than the second input. ≤: Return true if the first input is smaller than or equal to the second input. >: Return true if the first input is greater than the second input. ≥: Return true if the first input is greater than or equal to the second input.

2 

and: Return true if both inputs are true; **or:** Return true if at least one of the inputs is true

3 

Returns true if the input is false. Returns false if the input is true.

4 

Returns either true or false.

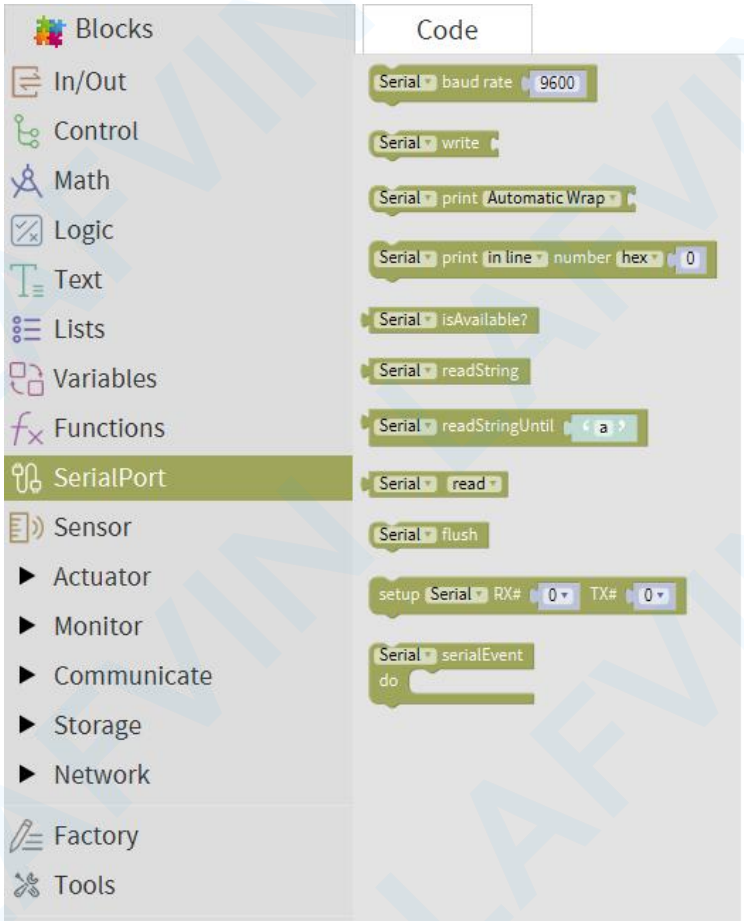
5 

Returns null

6 

If the first number is true, the second number is returned, otherwise the third number.

SerialPort Block:



NO.	BLOCK ICON	DEFINITION
1		Set the serial buad rate to 9600
2		Write the specified number, text or other value.
3		Print the specified number, text or other value on monitor.
4		Print the specified number, text or other value on newline of monitor.

5 

Print the specified number in hexademical format on newline of monitor.

6 

If the serial port is available, it returns true, otherwise returns false. (generally used in Bluetooth communication)

7 

Returns a string in serial port

8 

A string read from serial port to a string variable, pause until read the specified character.

9 

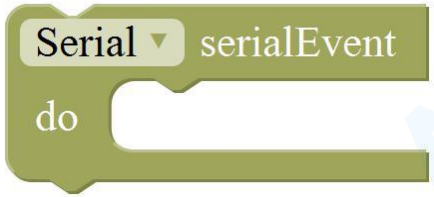
Read the serial data by byte (generally used to read the value sent from Bluetooth) (delete the data has been read)

10 

Wait for the output data completed

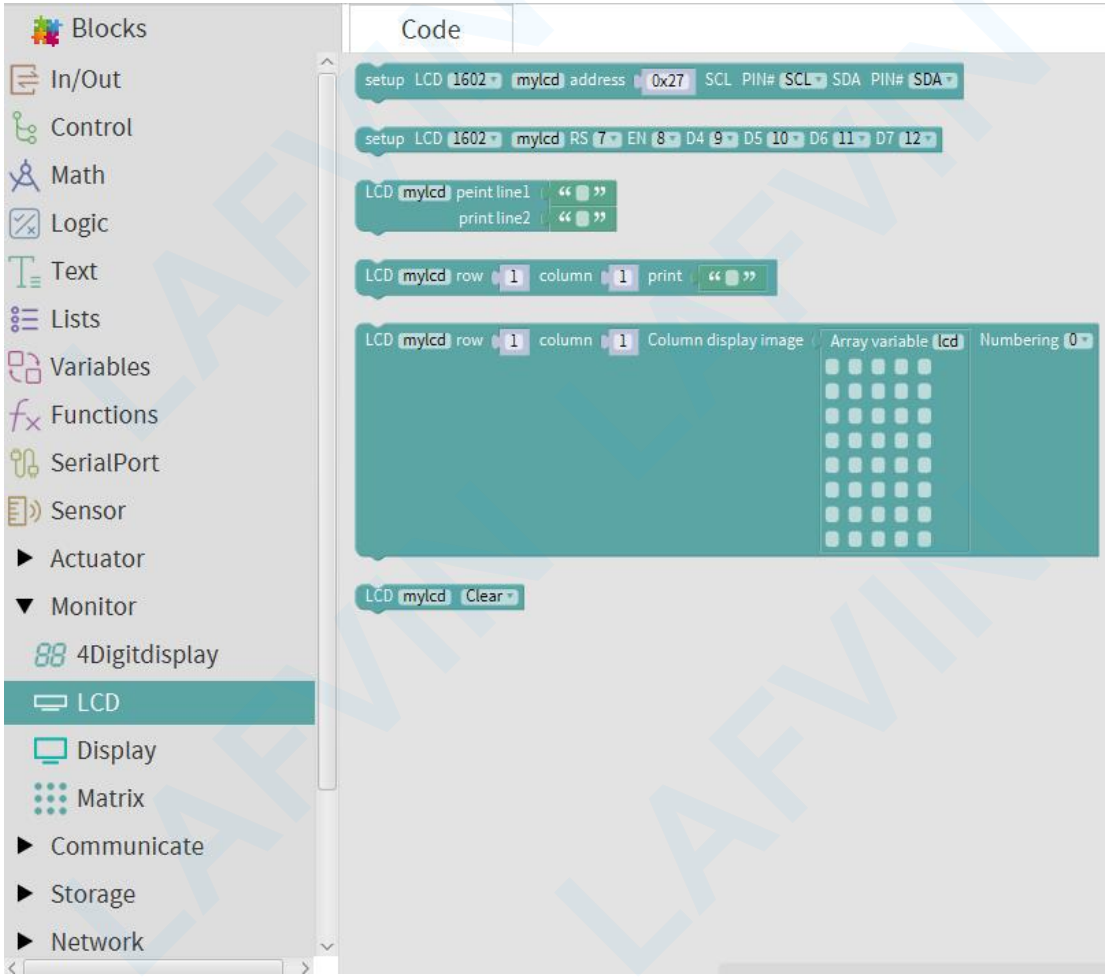
11 

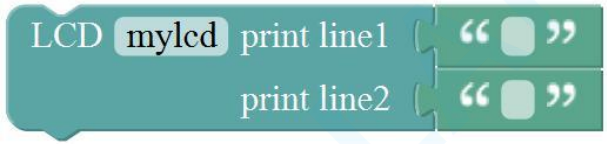
Set the software serial port (call this function if need to use several serial ports)

12 

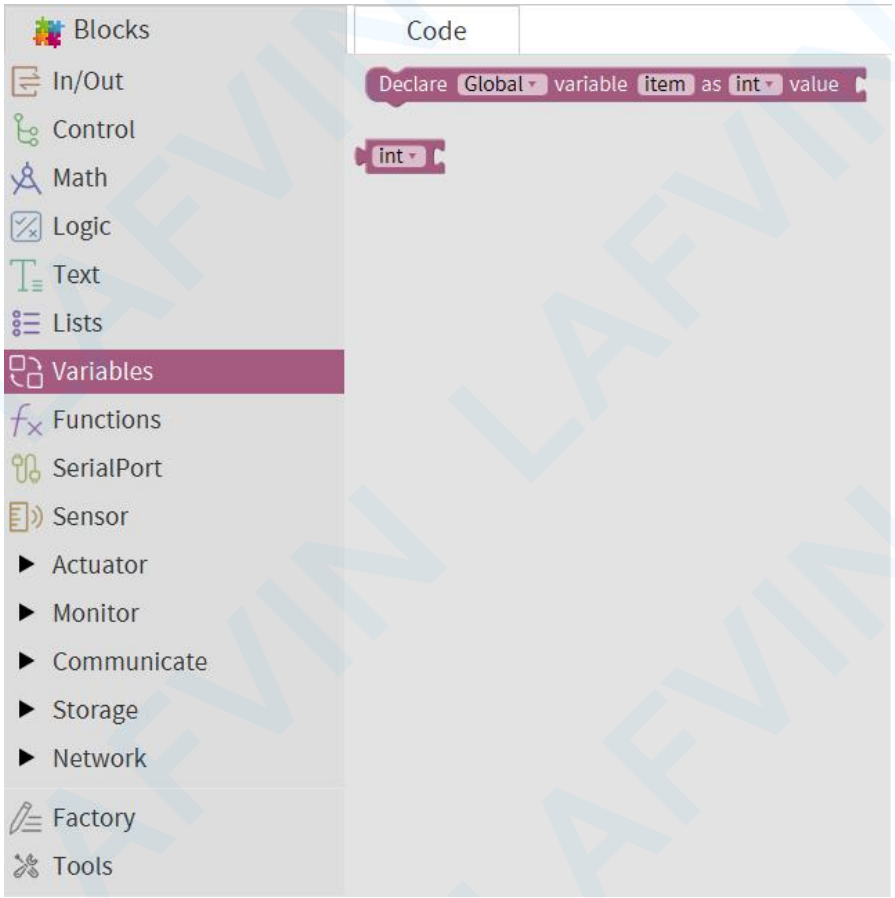
Event function trigger by serial port data, that is, serial port is ready to call this function. (equal to an interrupt function)

Monitor Block:



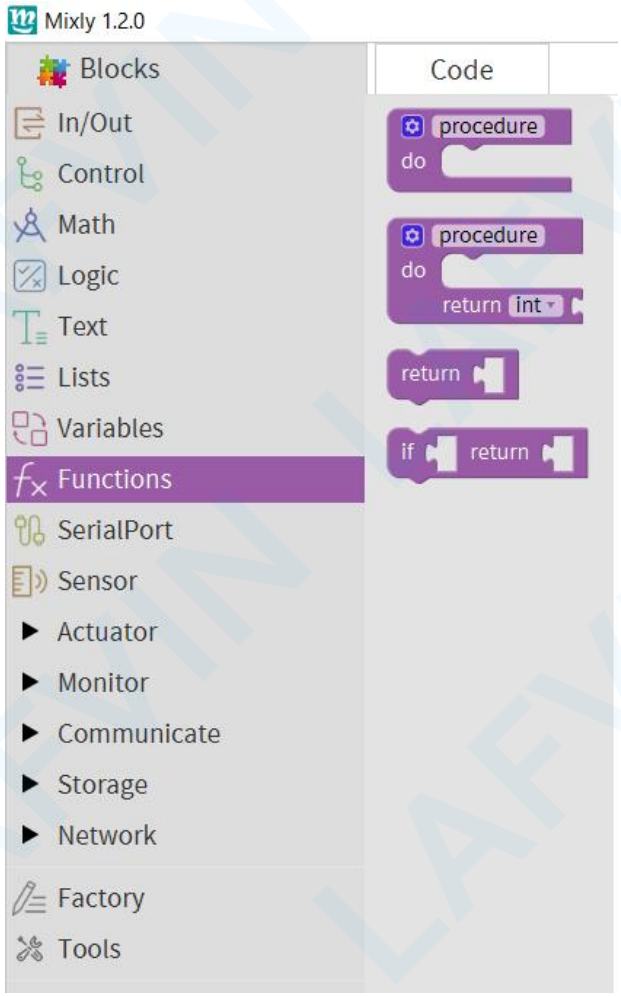
NO.	BLOCK ICON	DEFINITION
1		Set the IIC LCD1602 address
2		Input the value on LCD line 1 and line 2 from left to right.
3		Set the row and column of LCD to print the char
4		Clear the LCD screen

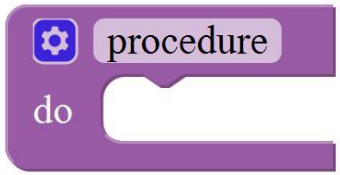
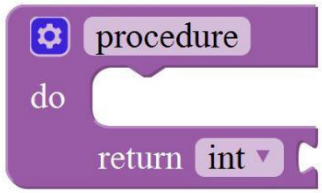
Variables Block



NO.	BLOCK ICON	DEFINITION
1		Define an integer variable whose name is item
2		Mandatory type conversion of constants or variables

Functions Block:



NO.	BLOCK ICON	DEFINITION
1		Creates a function with no output. Click the blue icon to set the procedure parameter. (no return value)
2		Creates a function with an output. Click the blue icon to set the procedure parameter. (with return value and can set the data types)
3		If a value is true, then return a second value.

4.Project

Alright, let's get straight to our projects. In this kit, there are 16 sensors and modules included. We will start with the simple sensor to make you know the smart home deeply.However, if you are an enthusiast with Arduino knowledge. You could skip theses steps, assemble the smart home kit directly(there is assembly video in the folder)

Note: In this course, the interface of each sensor / module marked with(G,-, GND) indicates the negative pole, G is connected to G or - or GND of sensor shield or control board; "V" is positive pole and linked with V or VCC or 5V.Since the Bluetooth module uses serial communication, uploading code to the Arduino uno main control board also needs serial communication. In order to avoid occupying the serial port at the same time, causing the upload code to fail, you need to disconnect the Bluetooth module from the V5 sensor expansion board before uploading the code.

Project 1：Sound Photosensitive Control LED

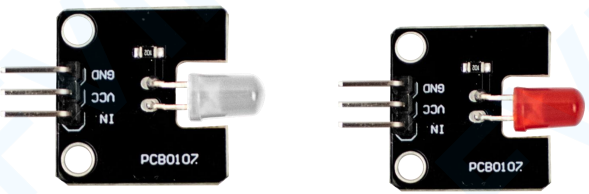
Overview

In ordinary homes, lighting lamps are controlled by manual switches, and manual switches are generally installed at a certain height from the ground. If there are elderly and children at home, it will be very inconvenient to use. If the lighting becomes automatically controlled by induction, will it be more intelligent and convenient? In this project, we will learn to make an LED lighting that can be controlled by sound and light intensity at the same time.

When it is in the daytime, the dark intensity detected by the photosensitive sensor is less than 500, and the light will not light up regardless of the sound detected by the sound sensor.

When it is dark, the light sensor detects that the intensity of darkness is greater than 500. If the sound sensor detects the sound of people walking nearby, the light will turn on.

LED Module



Specifications

Control interface: digital port

Working voltage: DC 3.3-5V

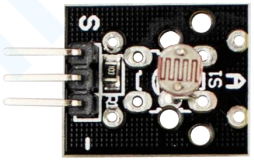
Pin pitch: 2.54mm

LED display color: white,red

Display color: white,red

Power on GND and VCC, the LED will light up when signal end S is high level, on the contrary, LED will turn off when signal end S is low level.

Photocell Sensor



The photoelectric sensor (photoresistor) is a resistor semiconductor made by the photoelectric effect. It is very sensitive to ambient light, so its resistance value changes with different light intensities. We use its functions to design circuits and generate photoresistor sensor modules. The signal end of the module is connected to the microcontroller.

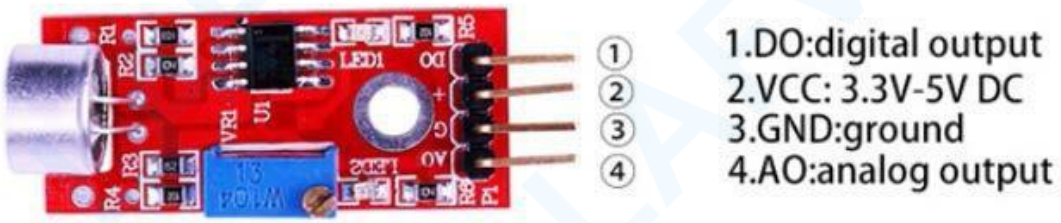
When the light intensity increases, the resistance decreases, and the voltage of the signal output port of the module decreases, that is, the voltage detected by the analog port of the microcontroller will decrease.

Otherwise, when the light intensity decreases, the resistance increases, and the voltage of the signal output port of the module increases, that is, the voltage detected by the analog port of the microcontroller will increase.

Therefore, we can use the photoresistor sensor module to read the corresponding analog value and sense the light intensity in the environment. It is usually used in light measurement, control and conversion, and light control circuits.

Sound sensor module

This module is used to judge the intensity of the sound. The intensity of the sound can be converted into the magnitude of the voltage signal. The module has two outputs:



AO: analog output, microphone real-time output voltage signal

DO: When the intensity of the sound reaches a certain threshold, the output is a high-level or low-level signal. Threshold sensitivity can be achieved by adjusting the potentiometer.

How to adjust the sensitivity of the sound module

Observe the status of the LED lights on the module.

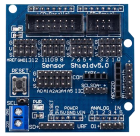
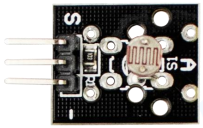
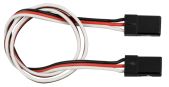
Case 1:

If LED1 is on, use a screwdriver to turn the blue potentiometer counterclockwise until the red LDE light on the module changes from on to off, stop the rotation and record this position A, position A is the most sensitive. On the basis of position A, rotate it counterclockwise slightly to reduce the sensitivity.

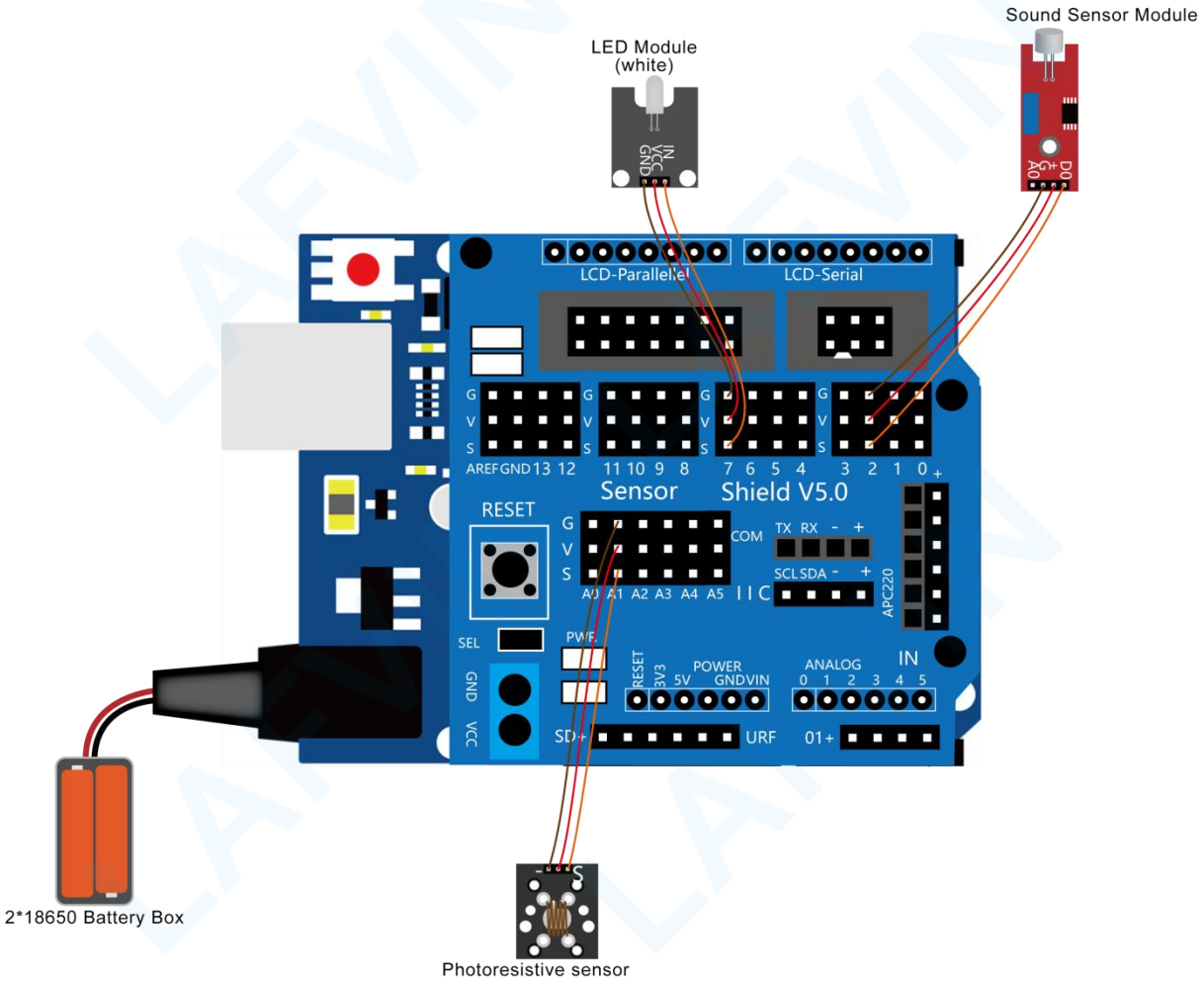
Case 2:

If LED1 is off, use a screwdriver to turn clockwise on the blue potentiometer until the red LDE light on the module changes from the off state to the on state, and then adjust according to the method in Case 1.

Experiment equipment：

Arduino UNO*1	Arduino Sensor Shield V5.0*1	USB cable*1	Photocell Sensor*1	Sound sensor module*1	LED module （white）*1	3pin F-F Dupont line*3
						

Wiring Diagram：



Let's program

Project purpose

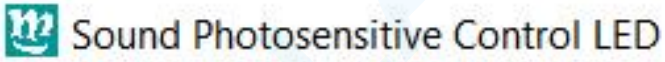
When it is in the daytime, the dark intensity detected by the photosensitive sensor is less than 500, and the light will not light up regardless of the sound detected by the sound sensor.

When it is dark, the light sensor detects that the intensity of darkness is greater than 500. If the sound sensor detects the sound of people walking nearby, the light will turn on.

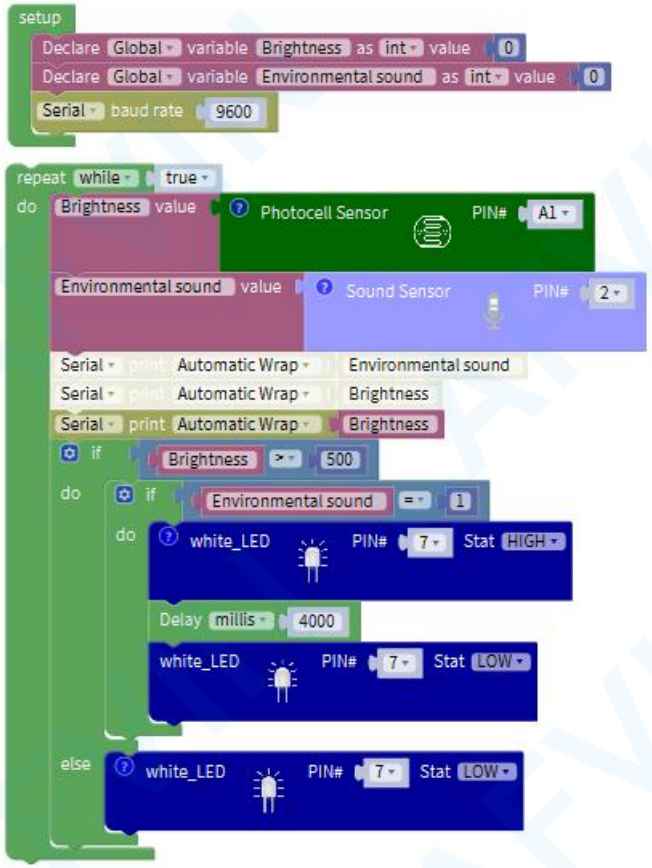
Note: you could set the range of analog value freely based on test result

Mixly Code

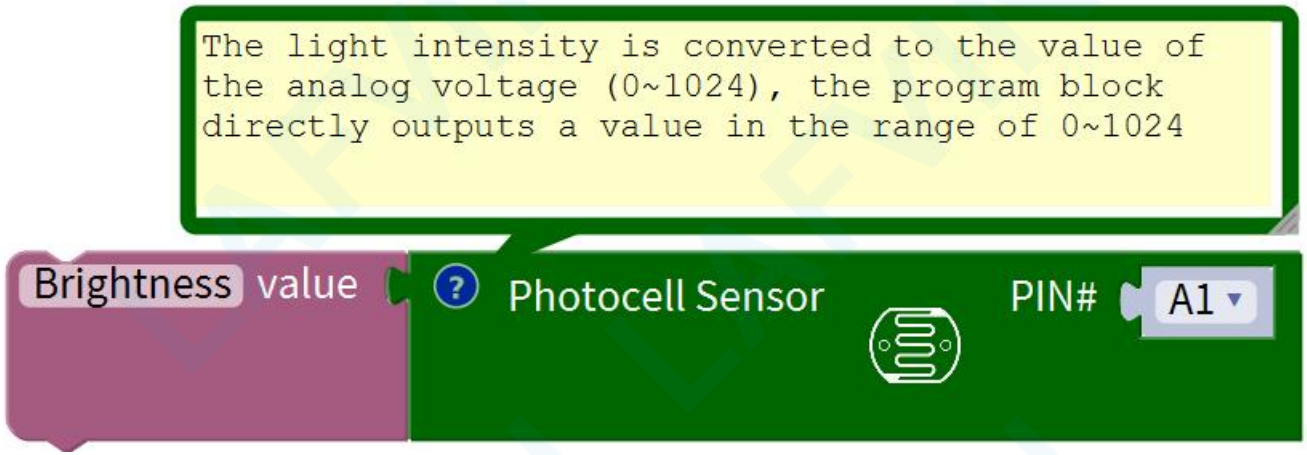
Wire it up well as the above diagram. Okay, let's move on to write the test code.Open mixly software.Click "**New**" to add new project, then start your programming .

if you want to refer to the program we provide.open the reference code for this project"**Sound Photosensitive Control LED.mix**" 

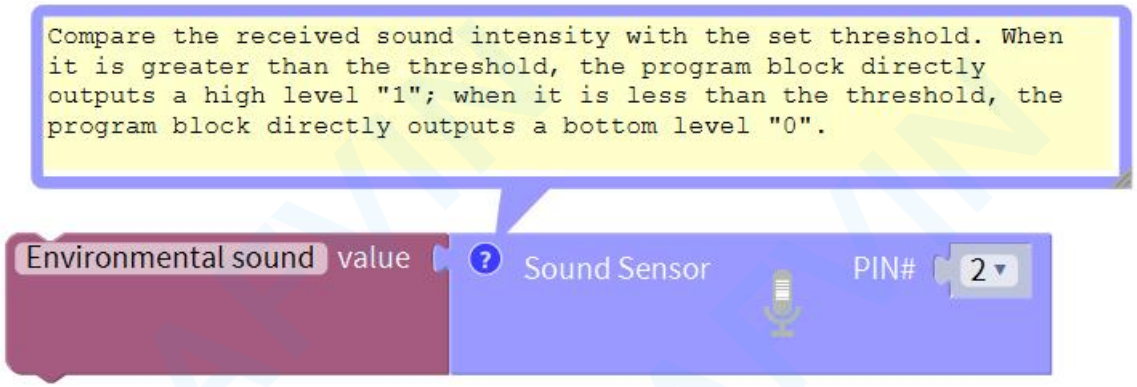
in the reference materials we provided.


setup
Declare Global variable Brightness as int value 0
Declare Global variable Environmental sound as int value 0
Serial baud rate 9600
repeat while true
do
Brightness value
Photocell Sensor
PIN# A1
Environmental sound value
Sound Sensor
PIN# 2
Serial print Automatic Wrap Environmental sound
Serial print Automatic Wrap Brightness
Serial print Automatic Wrap Brightness
if Brightness > 500
do
if Environmental sound = 1
do
white_LED
PIN# 7
Stat HIGH
Delay millis 4000
white_LED
PIN# 7
Stat LOW
else
white_LED
PIN# 7
Stat LOW

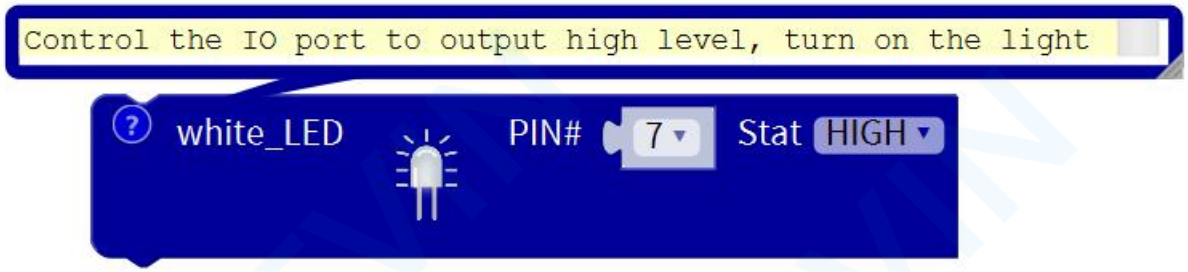
Programming Thinking



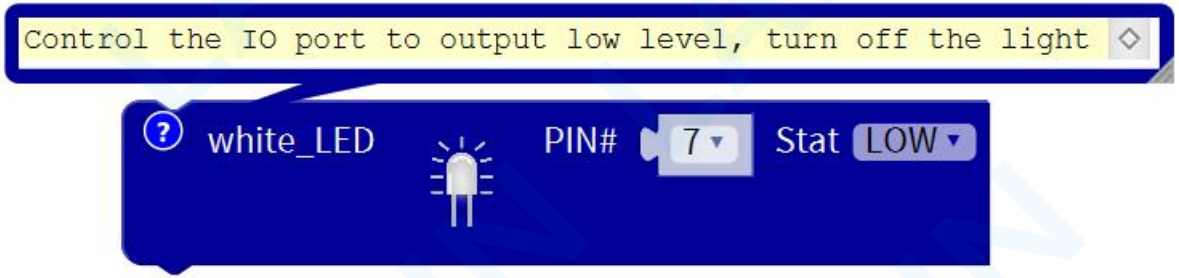
The photosensitive sensor is connected to the analog port A1. The photosensitive sensor converts the light intensity into an analog voltage value (0~1024), and the program block directly outputs a value in the range of 0~1024. Save the obtained value to the variable Brightness.



The sound sensor module is connected to the digital port D2 and compares the received sound intensity with the set threshold. When it is greater than the threshold, the program block directly outputs a high level "1"; when it is less than the threshold, the program block directly outputs the bottom voltage Level "0" and save the obtained level state to the variable Environmental sound

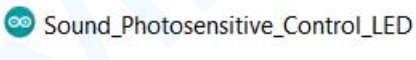


Connect the LED light module to the digital port D7, control the D7 port to output high level, turn on the light



Connect the LED light module to the digital port D7, control the D7 port to output low level, turn off the light

Arduino Code

If you want to refer to the program we provide.Open the reference code for this project "Sound_Photosensitive_Control_LED.ino" 

in the reference materials we provided.

Programming Thinking

```
/*
LAFVIN Smart Home Kit for Arduino
Project 1
*/
volatile int Brightness;
volatile int Environmental_sound;
void setup(){
   Brightness = 0;
```

```
    Environmental_sound = 0;
    Serial.begin(9600);
    pinMode(A1, INPUT);// initialize digital pin A1,D2 as an input.
    pinMode(2, INPUT);
    pinMode(7, OUTPUT);// initialize digital pin D7 as an output.
}

void loop()
{
      Brightness = analogRead(A1);//Read the output analog voltage value of the photosensitive sensor and save it to the variable Brightness
      Environmental_sound = digitalRead(2);//Read the digital output status of the sound sensor and save it to the variable Environmental sound
```

```
Serial.println(Brightness);
if (Brightness > 500)
{
   if (Environmental_sound == 1)
   {
      digitalWrite(7,HIGH);// turn the LED on (HIGH is the voltage level)
      delay(4000); // wait for 4 seconds
      digitalWrite(7,LOW);// turn the LED off by making the voltage LOW
   }
}
else
{
   digitalWrite(7,LOW);
```

```
    }
}
```

Test Result:

Upload the test code to UNO R3 control board, turn the POWER switch ON.

When it is in the daytime, the dark intensity detected by the photosensitive sensor is less than 500, and the light will not light up regardless of the sound detected by the sound sensor.

When it is dark, the light sensor detects that the intensity of darkness is greater than 500. If the sound sensor detects the sound of people walking nearby, the light will turn on and the LED will automatically turn off after 4 seconds.

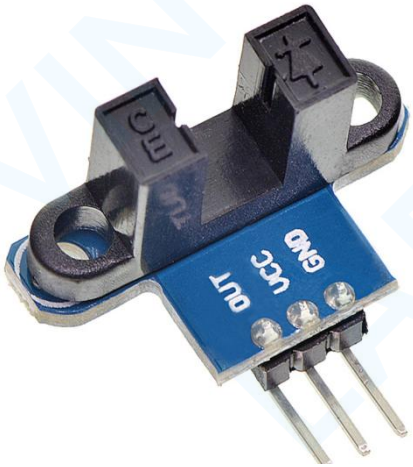
(Note: Before uploading the code, you need to disconnect the Bluetooth module from the sensor shield V5 After successfully uploading the code, reinstall the Bluetooth module.)

Project 2：Coin Parking

Overview

In the smart home, the garage is an essential part. In this project, we made a coin-operated parking device, using photoelectric speed sensor and servo. When a coin is inserted, the parking gate will open, and when the coin is taken out, the parking gate will be closed.

Photoelectric interrupt Module

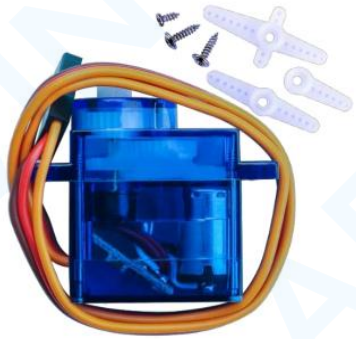


It adopts slot-type photoelectric sensor, which consists of an infrared light-emitting diode and an NPN phototransistor, and the slot width is 5.9mm. As long as the non-transparent object passes through the groove, the TTL low level can be triggered.Connect VCC and GND, the signal indicator of the module will be on.

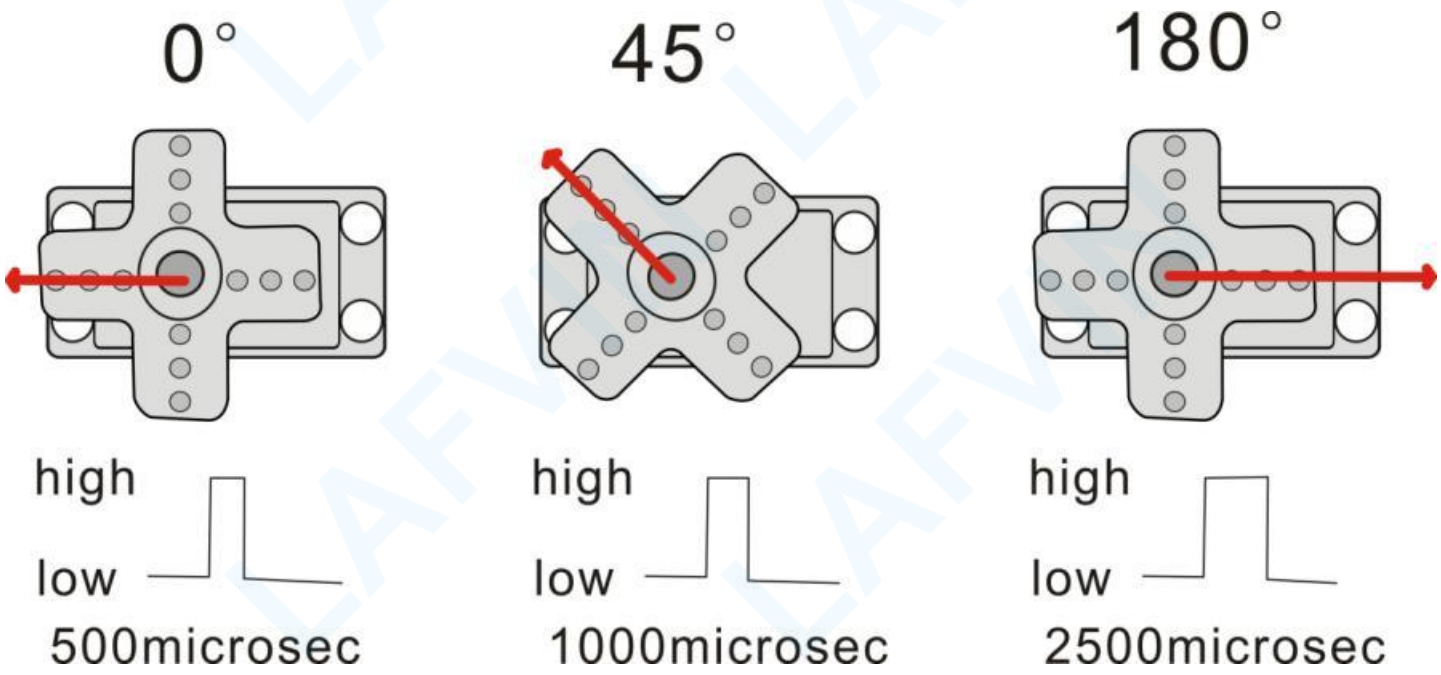
When there is no block in the module slot, the receiving tube is turned on and the module OUT outputs high level;

when blocked, OUT outputs low level and the signal indicator is off. The module OUT can be connected with a relay to form a limit switch and other functions, or it can be connected with an active buzzer module to form an alarm.

Servo

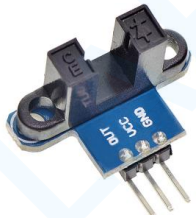
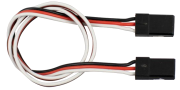


When we make this kit, we often control doors and windows with servos. In this course, we'll introduce its principle and how to use servo motors.Servo motor is a position control rotary actuator. It mainly consists of housing,circuit board, core-less motor, gear and position sensor. Its working principle is that the servo receives the signal sent by MCU or receiver and produces a reference signal with a period of 20ms and width of 1.5ms, then compares the acquired DC bias voltage to the voltage of the potentiometer and obtain the voltage difference output.

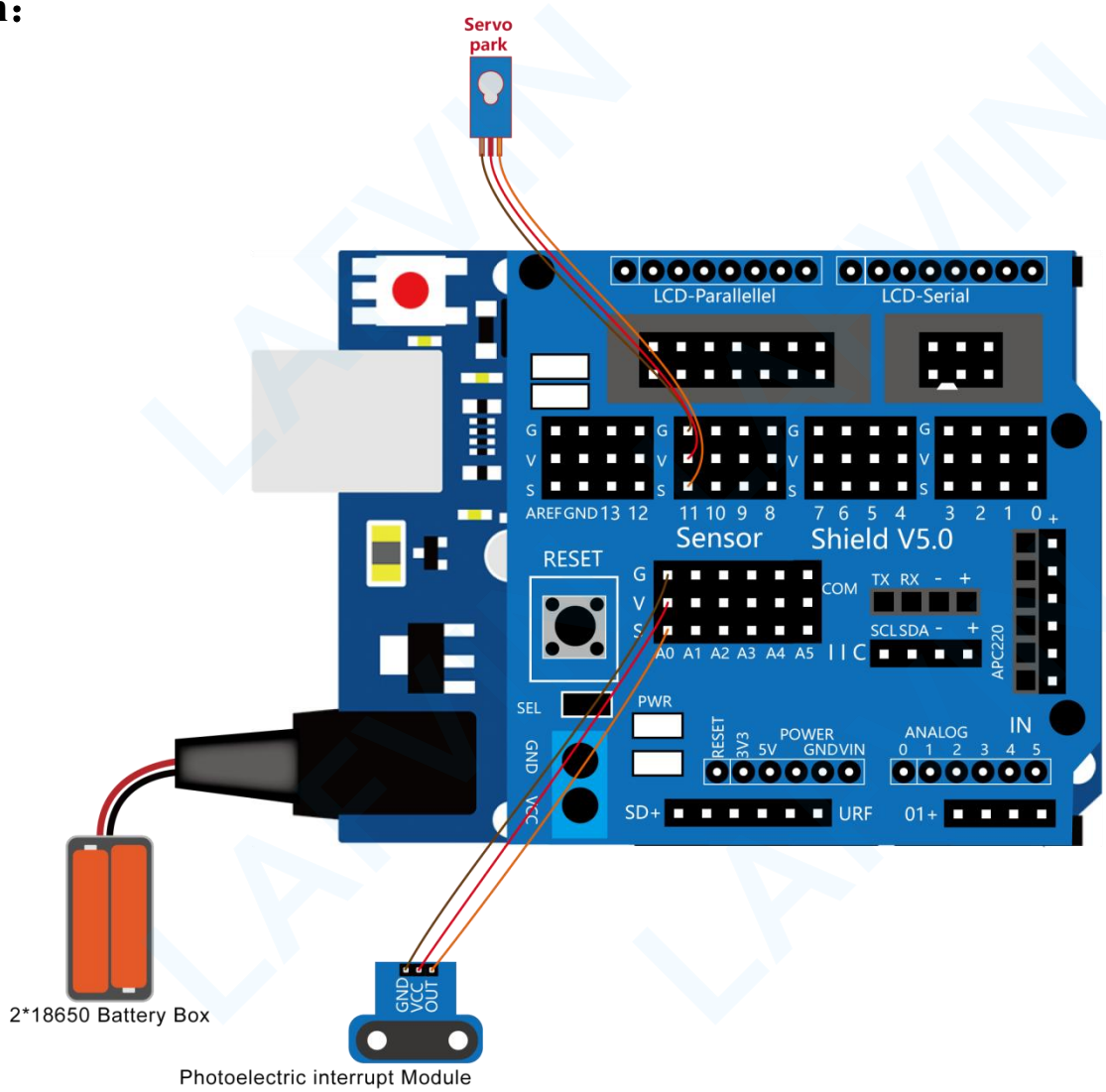


When the motor speed is constant, the potentiometer is driven to rotate through the cascade reduction gear, which leads that the voltage difference is 0, and the motor stops rotating. Generally, the angle range of servo rotation is 0° --180 ° The rotation angle of servo motor is controlled by regulating the duty cycle of PWM (Pulse-Width Modulation) signal. The standard cycle of PWM signal is 20ms(50Hz). Theoretically, the width is distributed between 1ms-2ms, but in fact, it's between 0.5ms-2.5ms. The width corresponds the rotation angle from 0° to 180°.But note that for different brand motor, the same signal may have different rotation angle.One is to use a common digital sensor port of Arduino to produce square wave with different duty cycle to simulate PWM signal and use that signal to control the positioning of the motor.

Experiment equipment:

Arduino UNO*1	Arduino Sensor Shield V5.0*1	USB cable*1	Photoelectric Speed Sensor *1	Servo*1	3pin F-F Dupont line*1
					

Wiring Diagram：



Let's program

Project purpose

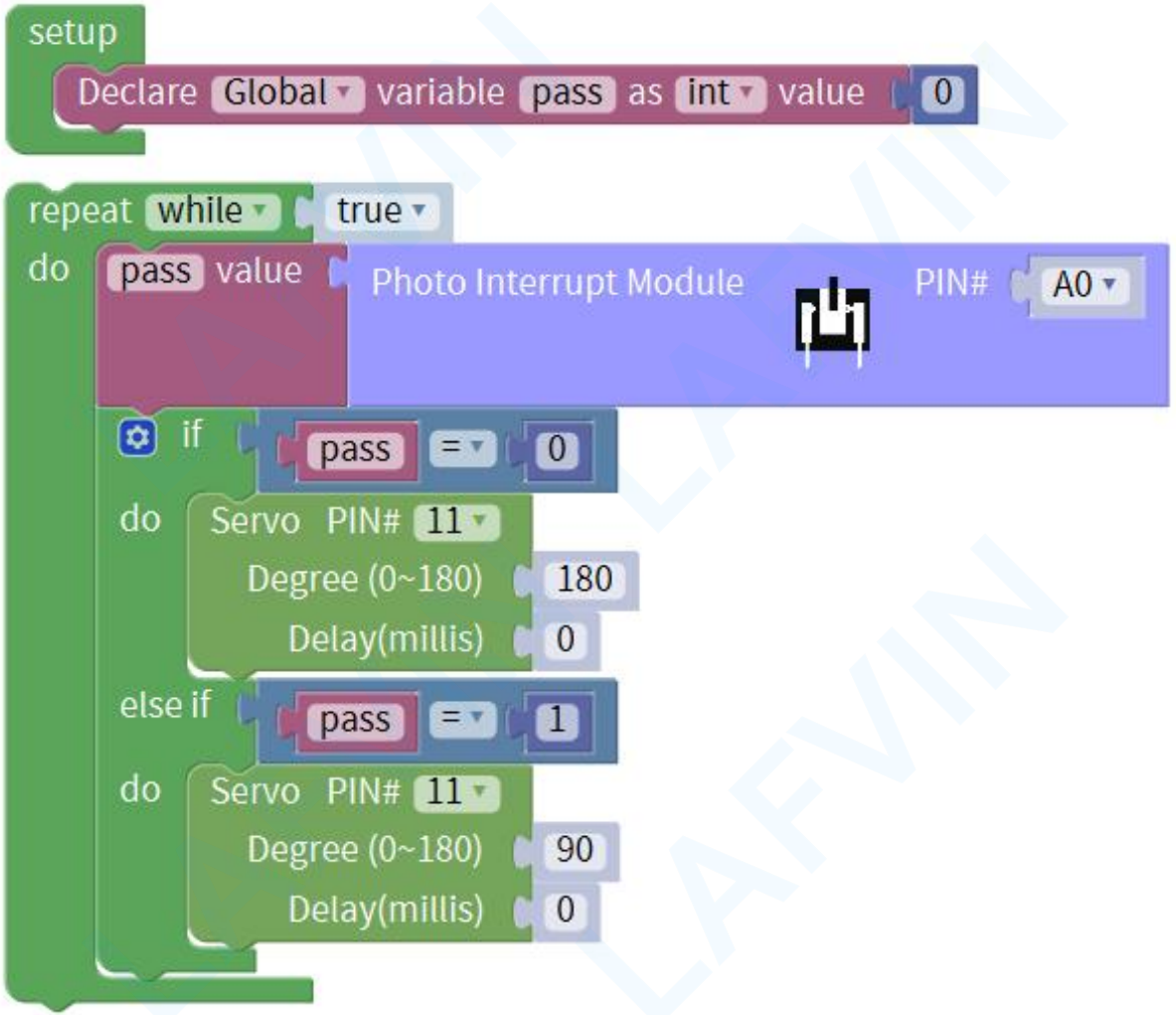
When a coin is inserted, the parking gate will open.

when the coin is taken out, the parking gate will be closed.

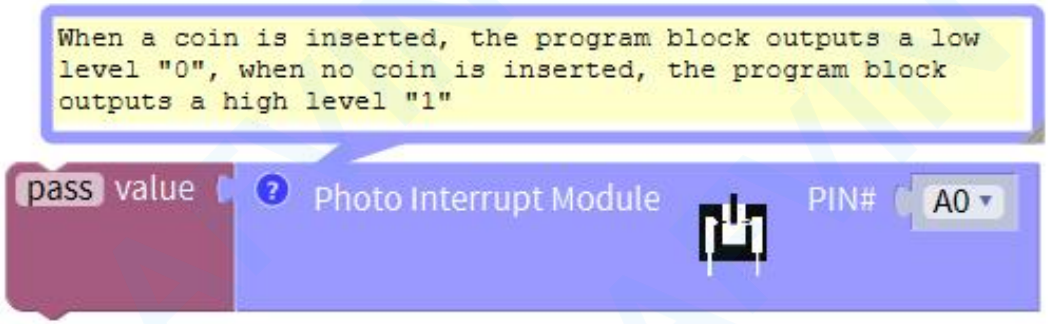
Mixly Code

Wire it up well as the above diagram. Okay, let’s move on to write the test code.Open Mixly software.Click “**New**” to add new project, then start your programming .

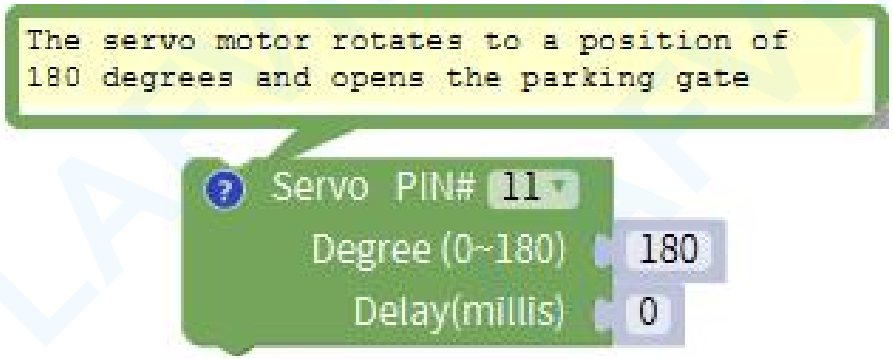
if you want to refer to the program we provide.open the reference code for this project"**Coin Parking.mix**" in the reference materials we provided.


setup
Declare Global variable pass as int value 0
repeat while true
do
pass value
Photo Interrupt Module
PIN# A0
if pass = 0
do
Servo PIN# 11
Degree (0~180) 180
Delay(millis) 0
else if pass = 1
do
Servo PIN# 11
Degree (0~180) 90
Delay(millis) 0

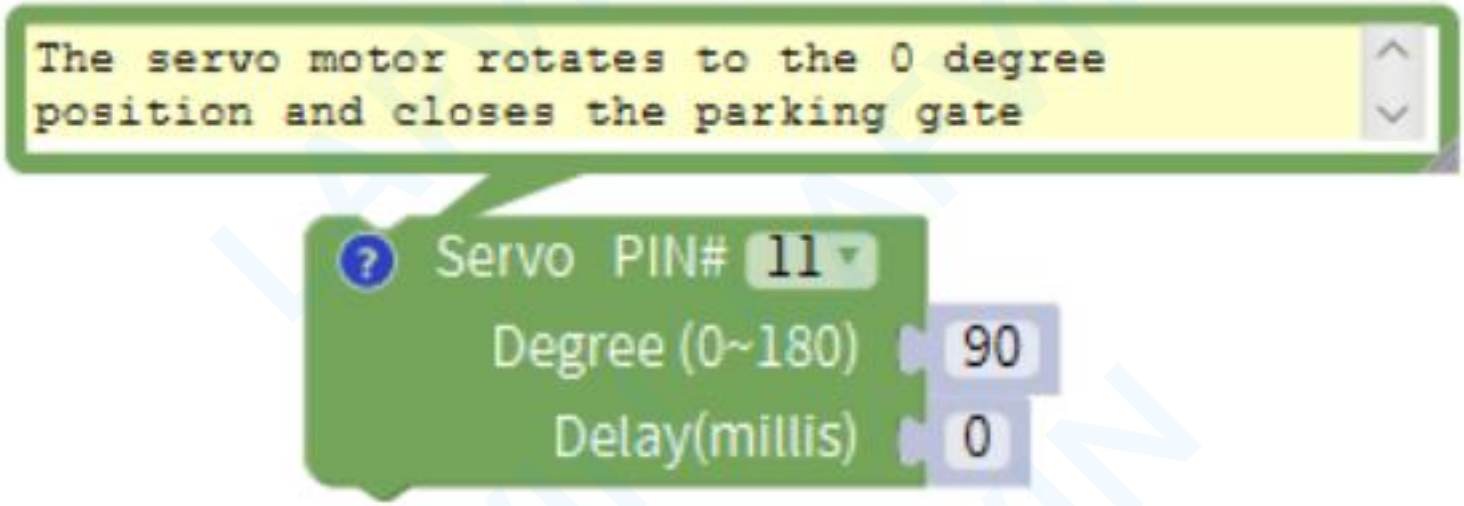
Programming Thinking



Photoelectric interrupt Module is connected to the analog port A0. When a coin is inserted, the program block outputs a low level "0", when no coin is inserted, the program block outputs a high level "1", and the obtained value is saved to the variable pass.



The servo motor that controls the parking gate is connected to the digital port D11. The program block commands to control the servo motor to rotate to a position of 180 degrees and open the parking gate.



The servo motor that controls the parking gate is connected to the digital port D11, the program block commands to control the servo motor to rotate to the 90 degree position and close the parking gate

Arduino Code

If you want to refer to the program we provide.Open the reference code for this project "**Coin Parking.ino**" in the reference materials we provided.

Before you can run this, make sure that you have installed the < Servo> library or re-install it, if necessary. Otherwise, your code won't work.For details about loading the library file, see Lesson**"2.3 How to Add Libraries"** about how to add libraries .

Programming Thinking

```
/*
LAFVIN Smart Home Kit for Arduino
Project 2
*/


#include <Servo.h>// Servo function library
```

```
int pass;
Servo servo_11;

void setup()
{
    pass = 0;
    pinMode(A0, INPUT);
    servo_11.attach(11);// Define the position of the servo on D11
}

void loop()
{
```

```
    pass = digitalRead(A0);//Read the level status of the analog port A0, high level "1" means no coin input, low level "0" means coin input
    if (pass == 0)
    {
      servo_11.write(180);//servo motor rotate to a position of 180 degrees and open the parking gate.
      delay(0);
    }
    else if (pass == 1)
    {
      servo_11.write(90);//servo motor rotate to the 0 degree position and close the parking gate
      delay(0);
    }
}
```

Test Result:

Upload the test code to UNO R3 control board, turn the POWER switch ON.

When a coin is inserted, the parking gate will open.

when the coin is taken out, the parking gate will be closed.

(Note: Before uploading the code, you need to disconnect the Bluetooth module from the sensor shield v5. After successfully uploading the code, reinstall the Bluetooth module.)

Project 3：Rain-Controlled Window

Overview

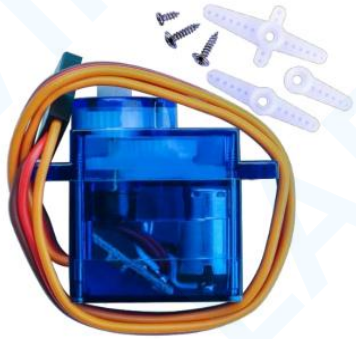
When it is windy and rainy, no one closes the windows at home, can you also make a rain-controlled window, so that you don’t have to worry about no one at home closing the windows, rainwater will enter the room, causing indoor humidity and other trouble.

The automatic smart window in this course is realized by using raindrop sensor and steering gear as the mechanism. The principle is that when the raindrop sensor senses rain, the steering gear will swing to close the window.

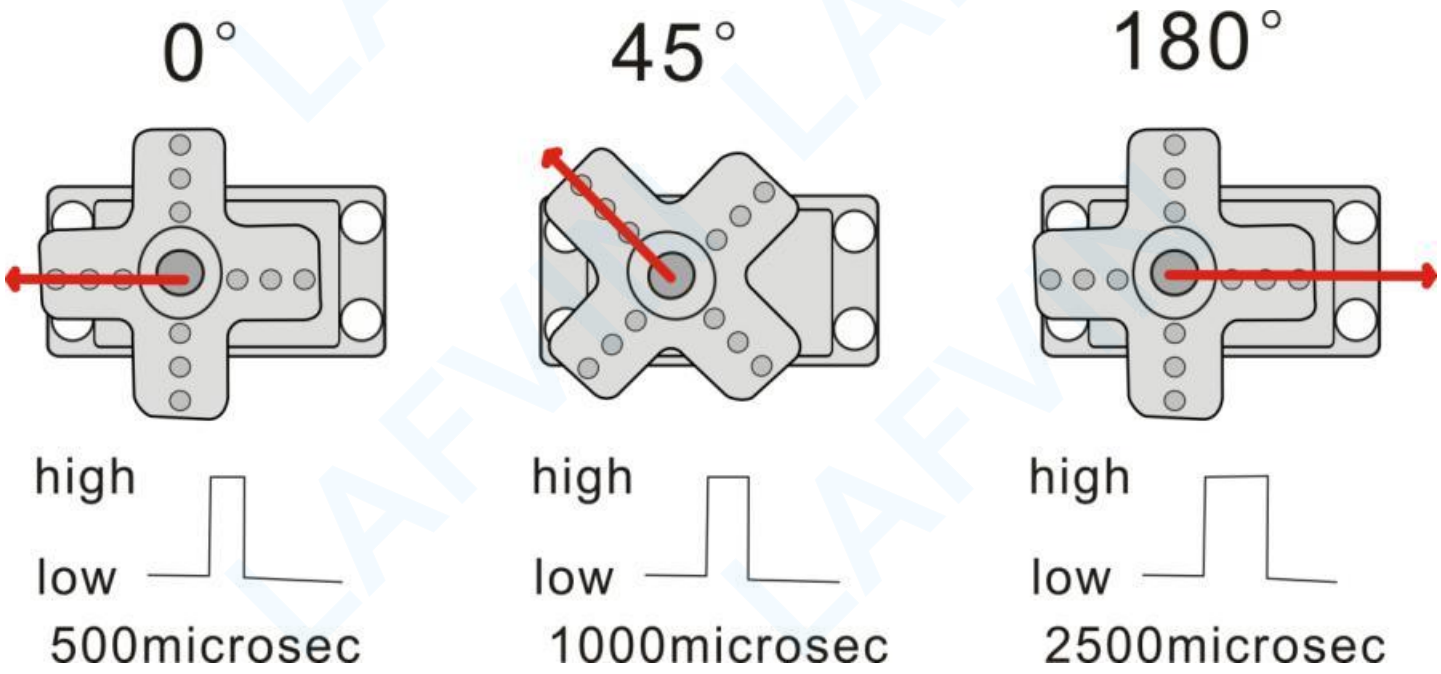
Water Lever Detection Sensor Module

Its principle is to detect the amount of water by bare printed parallel lines on the circuit board. The more the water is,the more wires will be connected. As the conductive contact area increases, the output voltage will gradually rise. It can detect water vapor in the air as well. The steam sensor can be used as a rain water detector and level switch. When the humidity on the sensor surface surges, the output voltage will increase.The sensor is compatible with various Arduino control boards, such as Arduino series Arduino control boards. When using it, we provide the guide to operate steam sensor and Arduino control board. Connect the signal end of the sensor to the analog port of the Arduino control boards, sense the change of the analog value, and display the corresponding analog value on the serial monitor.

Servo

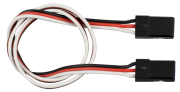


When we make this kit, we often control doors and windows with servos. In this course, we'll introduce its principle and how to use servo motors.Servo motor is a position control rotary actuator. It mainly consists of housing,circuit board, core-less motor, gear and position sensor. Its working principle is that the servo receives the signal sent by MCU or receiver and produces a reference signal with a period of 20ms and width of 1.5ms, then compares the acquired DC bias voltage to the voltage of the potentiometer and obtain the voltage difference output.

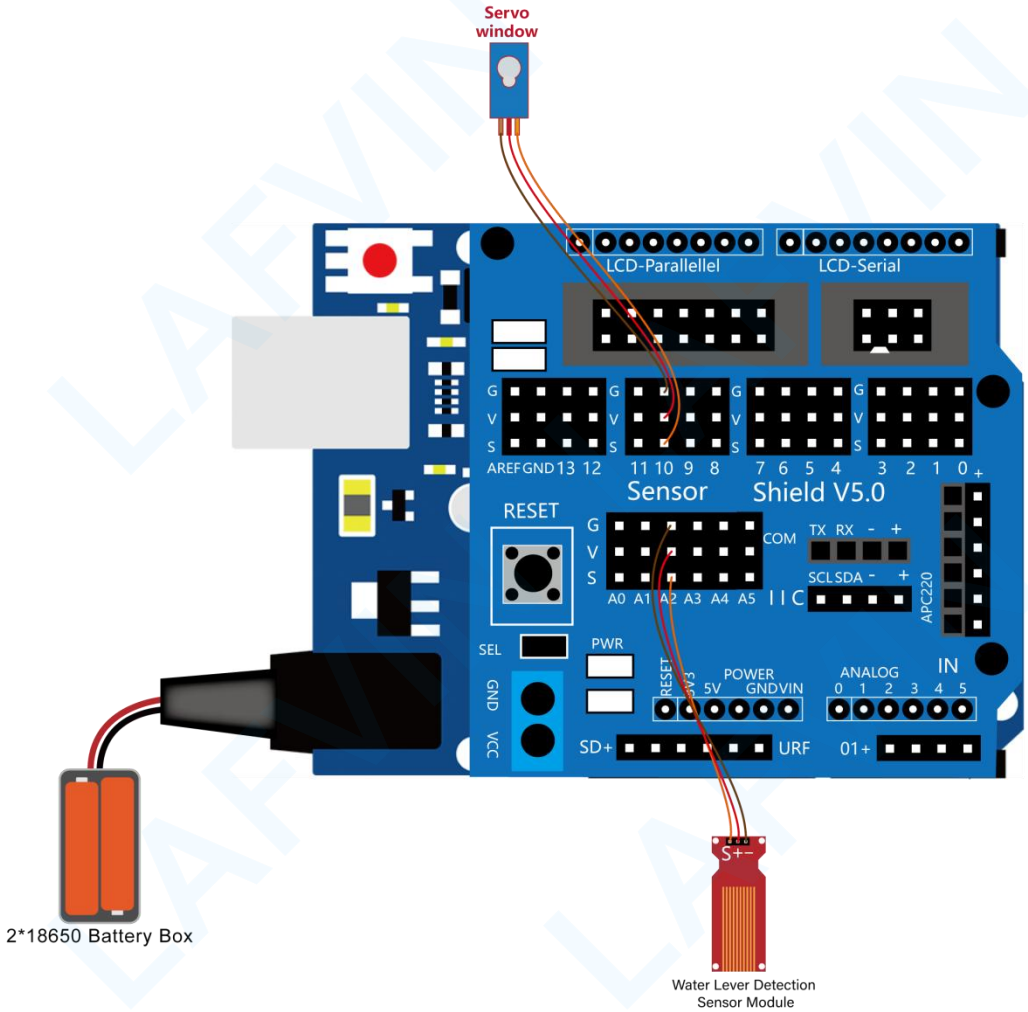


When the motor speed is constant, the potentiometer is driven to rotate through the cascade reduction gear, which leads that the voltage difference is 0, and the motor stops rotating. Generally, the angle range of servo rotation is 0° --180 ° The rotation angle of servo motor is controlled by regulating the duty cycle of PWM (Pulse-Width Modulation) signal. The standard cycle of PWM signal is 20ms(50Hz). Theoretically, the width is distributed between 1ms-2ms, but in fact, it's between 0.5ms-2.5ms. The width corresponds the rotation angle from 0° to 180°.But note that for different brand motor, the same signal may have different rotation angle.One is to use a common digital sensor port of Arduino to produce square wave with different duty cycle to simulate PWM signal and use that signal to control the positioning of the motor.

Experiment equipment:

Arduino UNO*1	Arduino Sensor Shield V5.0*1	USB cable*1	Water Lever Detection Sensor Module*1	Servo*1	3pin F-F Dupont line*1
					

Wiring Diagram：



Let's program

Project purpose

When it rains, the sensor can directly detect the change of rainwater.

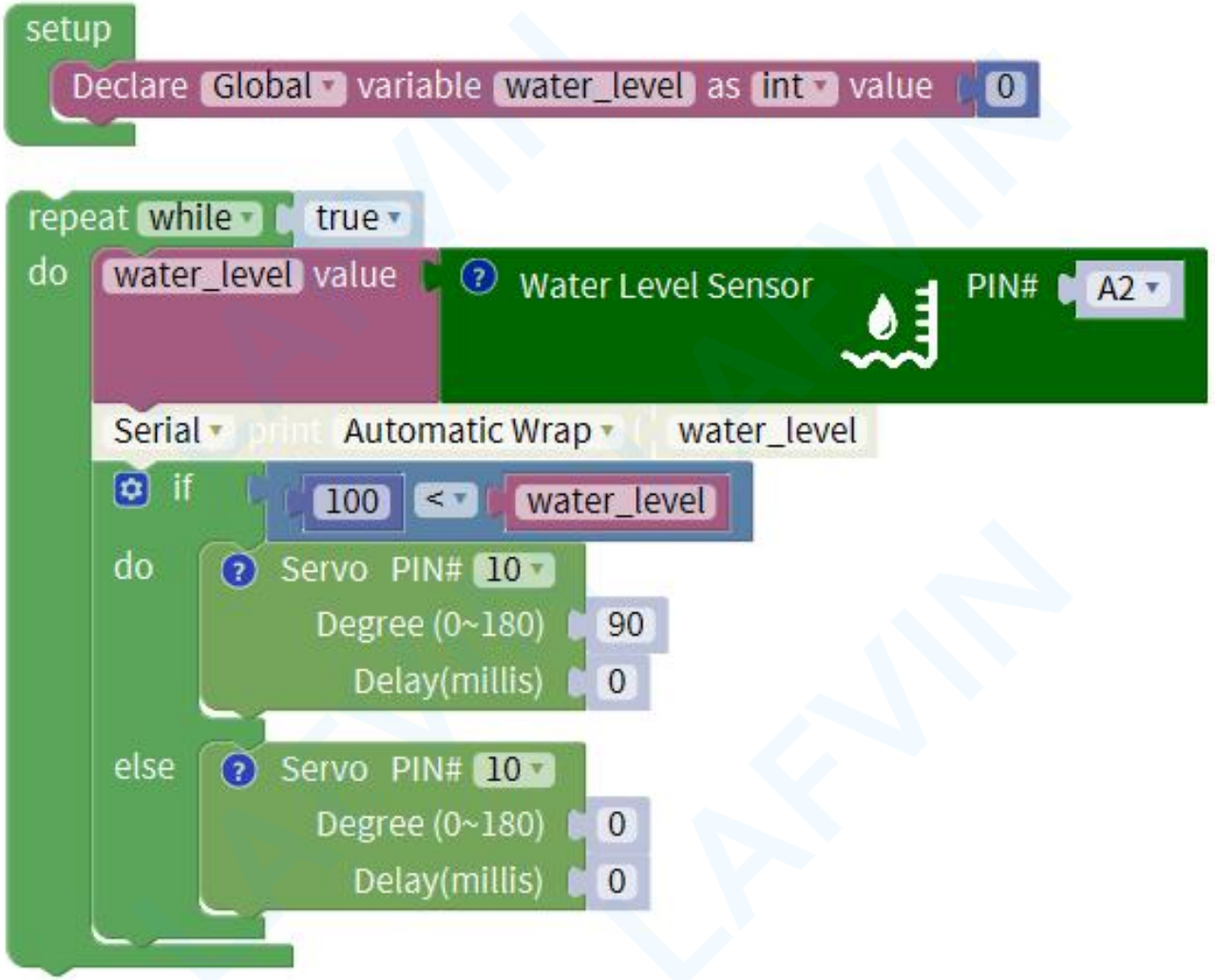
When it is raining, the water level detection sensor detects that the corresponding analog voltage value of the rainfall is greater than 100 and the window will be closed.

Otherwise the window is open

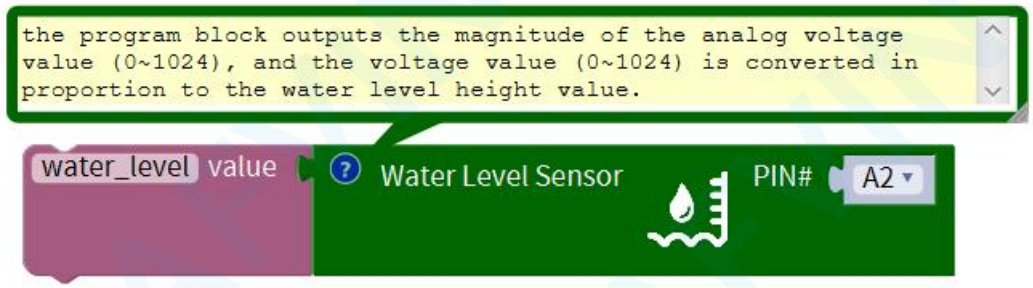
Mixly Code

Wire it up well as the above diagram. Okay, let’s move on to write the test code.Open Mixly software.Click “**New**” to add new project, then start your programming .

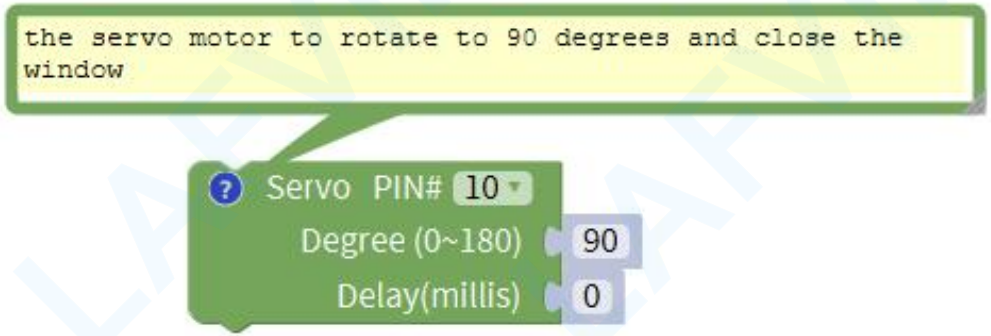
if you want to refer to the program we provide.open the reference code for this project"**Rain-Controlled Window.mix**" in the reference materials we provided.


setup
Declare Global variable water_level as int value 0
repeat while true
do
water_level value
Water Level Sensor
PIN# A2
Serial print Automatic Wrap water_level
if 100 < water_level
do
Servo PIN# 10
Degree (0~180) 90
Delay(millis) 0
else
Servo PIN# 10
Degree (0~180) 0
Delay(millis) 0

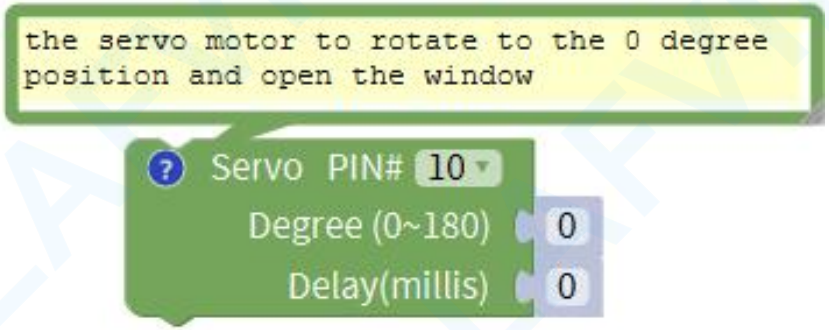
Programming Thinking



Photoelectric interrupt Module is connected to the analog port A0. When a coin is inserted, the program block outputs a low level "0", when no coin is inserted, the program block outputs a high level "1", and the obtained value is saved to the variable pass.



The servo motor that controls the window is connected to the digital port D10. This program block controls the servo motor to rotate to 90 degrees and close the window



The servo motor that controls the window is connected to the digital port D10. This program block controls the servo motor to rotate to the 0 degree position and open the window

Arduino Code

If you want to refer to the program we provide.Open the reference code for this project "**Rain-Controlled Window.ino**" in the reference materials we provided.

Programming Thinking

```
/*
LAFVIN Smart Home Kit for Arduino
Project 3
*/
#include <Servo.h>

int water_level;
Servo servo_10;

void setup()
{
    water_level = 0;
    pinMode(A2, INPUT);
```

```
    servo_10.attach(10);// Define the position of the servo on D10
}

void loop(){
    // the program block outputs the magnitude of the analog voltage value (0~1024), and the voltage value (0~1024) is converted in proportion to the water level height value.
    water_level = analogRead(A2);
    if (100 < water_level)
    {
      // the servo motor to rotate to 90 degrees and close the window
      servo_10.write(90);
    }
    else
```

```
    {
      // the servo motor to rotate to the 0 degree position and open the window
      servo_10.write(0);
    }
}
```

Test Result：

After uploading the program, the initial state of the window is open. The water level detection sensor is installed on the top of the house. When it rains, the sensor can directly detect the change of rainwater.

When it is raining, the water level detection sensor detects that the corresponding analog voltage value of the rainfall is greater than 100 and the window will be closed.

Otherwise the window is open

(Note: Before uploading the code, you need to disconnect the Bluetooth module from the sensor shield v5. After successfully uploading the code, reinstall the Bluetooth module.)

Project 4： Plant Watering Warning

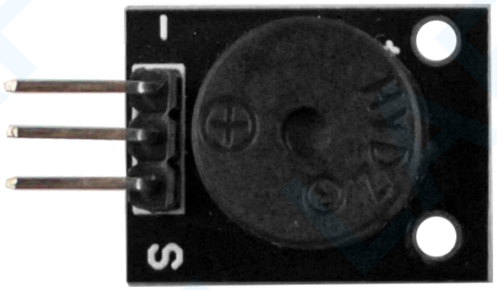
Overview

The significance of smart homes is to reduce the workload of humans and automate everything as much as possible. There are often potted plants in the home. If a warning device can be made to remind plants that need to be watered, it is important to prevent plants from drying out due to lack of water. This will use a sensor module that detects soil moisture-Soil Humidity Sensor.

Soil Humidity Sensor

This is a simple soil humidity sensor aims to detect the soil humidity. If the soil is in lack of water, the analog value output by the sensor will decrease; otherwise, it will increase. If you use this sensor to make an automatic watering device, it can detect whether your botany is thirsty to prevent it from withering when you go out.Using the sensor with Arduino controller makes your plant more comfortable and your garden smarter. The soil humidity sensor module is not as complicated as you might think, and if you need to detect the soil in your project, it will be your best choice.The sensor is set with two probes, when inserted into the soil, the sensor will get resistance value by reading the current changes between the two probes and convert such resistance value into moisture content. The higher moisture (less resistance), the higher conductivity the soil has.Its service life extends by metallizing the surface,Insert it into the soil and then use the AD converter to read it. With the help of this sensor, the plant can remind of you: I need water. It comes with 2 positioning holes for installing on other devices.

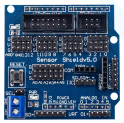
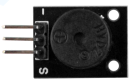
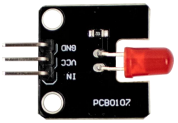
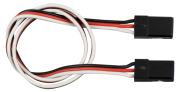
Passive Buzzer



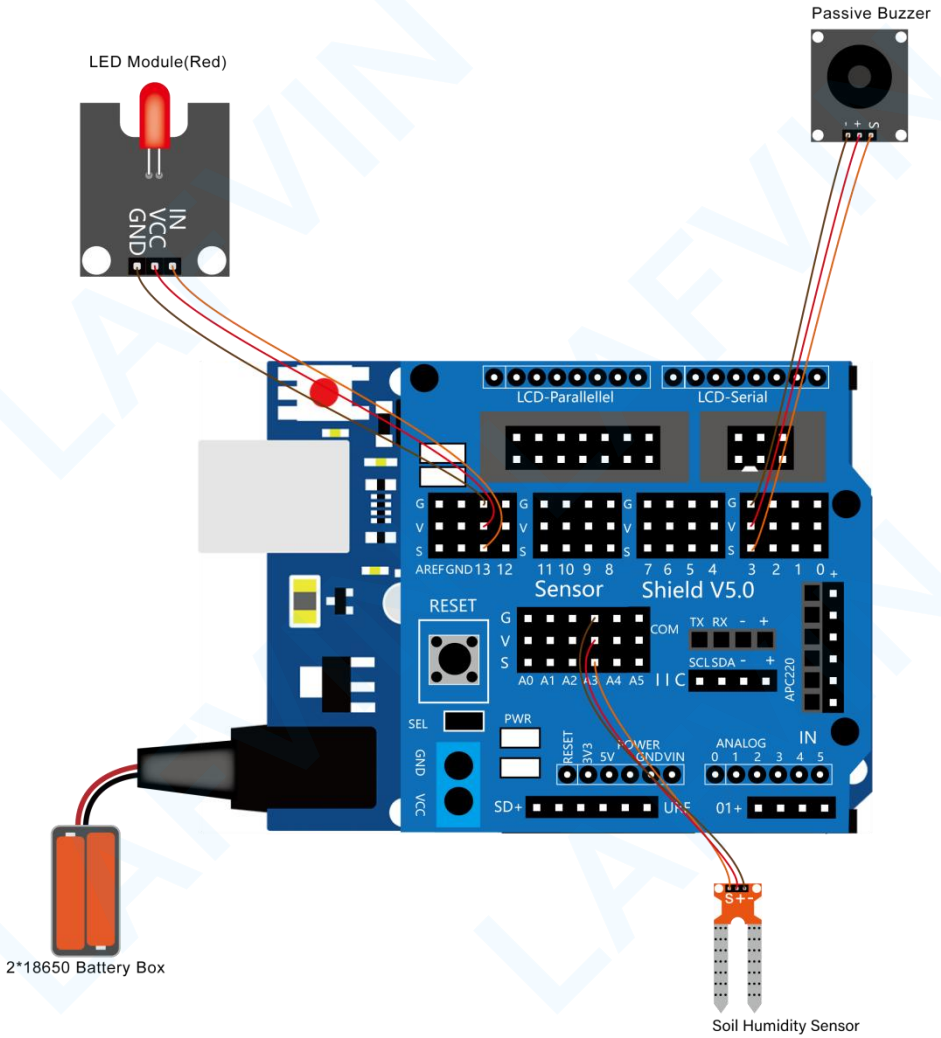
There are prolific interactive works completed by Arduino. The most common one is sound and light display. We always use LED to make experiments. For this lesson, we design circuit to emit sound. The universal sound components are buzzer and horns. Buzzer is easier to use. And buzzer includes about active buzzer and passive buzzer. In this experiment, we adopt passive buzzer.Use passive buzzer combined with red led light module to realize an audible and visual alarm. When the soil

moisture sensor detects that the soil is in a dry state, watering is required. The red LED of the audible and visual alarm will flash and the buzzer will Make a ticking sound.

Experiment equipment：

Arduino UNO*1	Arduino Sensor Shield V5.0*1	USB cable*1	Soil Humidity Sensor*1	Passive Buzzer*1	LED Module(red)*1	3pin F-F Dupont line*1
						

Wiring Diagram：



Let's program

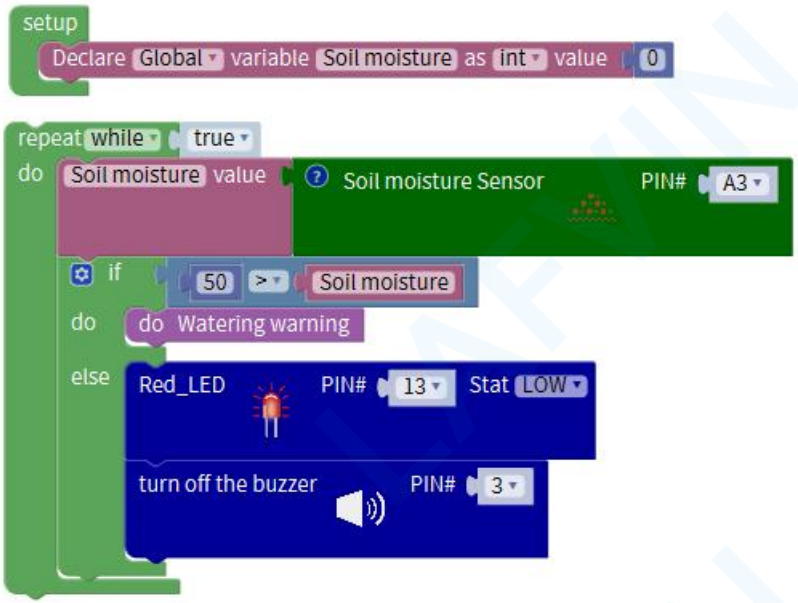
Project purpose

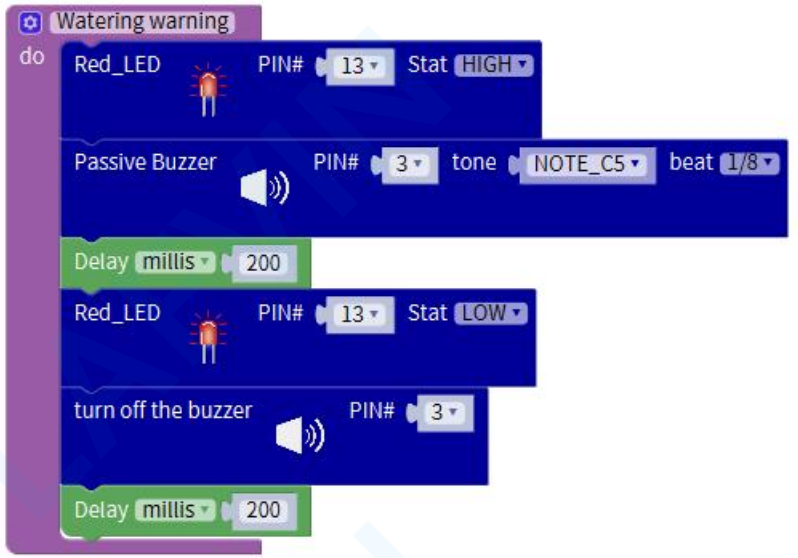
When the soil moisture sensor detects that the soil moisture value is less than 50, the sound and light alarm is activated, the red LED light of the sound and light alarm will flash, and the buzzer will emit a ticking sound.

Mixly Code

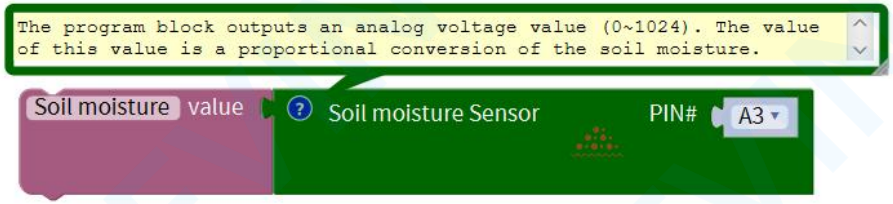
Wire it up well as the above diagram. Okay, let’s move on to write the test code.Open Mixly software.Click “**New**” to add new project, then start your programming .

if you want to refer to the program we provide.open the reference code for this project"**Plant Watering Warning.mix**" in the reference materials we provided.

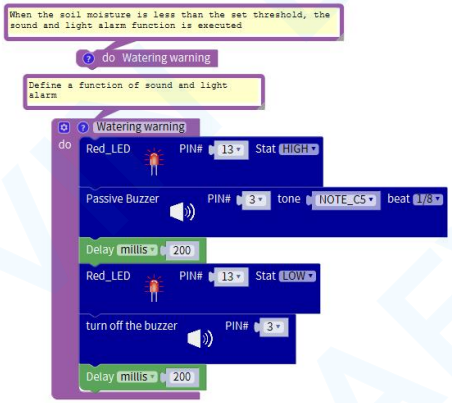

setup
Declare Global variable Soil moisture as int value 0
repeat while true
do
Soil moisture value
Soil moisture Sensor
PIN# A3
if 50 > Soil moisture
do
do Watering warning
else
Red_LED
PIN# 13
Stat LOW
turn off the buzzer
PIN# 3


Watering warning
do
Red_LED
PIN# 13
Stat HIGH
Passive Buzzer
PIN# 3
tone NOTE_C5
beat 1/8
Delay millis 200
Red_LED
PIN# 13
Stat LOW
turn off the buzzer
PIN# 3
Delay millis 200

Programming Thinking



The soil moisture sensor is connected to the analog port A3. The program block outputs an analog voltage value (0~1024). The value of this value is a proportional conversion of the soil moisture. Save the output value to the variable Soil moisture



The servo motor that controls the window is connected to the digital port D10. This program block controls the servo motor to rotate to 90 degrees and close the window

Arduino Code

If you want to refer to the program we provide.Open the reference code for this project "**Plant_Watering_Warning.ino**" in the reference materials we provided.

Programming Thinking

```
/*
LAFVIN Smart Home Kit for Arduino
Project 4
*/
int Soil_moisture;
// Define a function of sound and light alarm
void Watering_warning()
{
```

```
    digitalWrite(13,HIGH);
    tone(3,532);
    delay(125);
    delay(200);
    digitalWrite(13,LOW);
    noTone(3);
    delay(200);
}
void setup(){
    Soil_moisture = 0;
    pinMode(13, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(A3, INPUT);
```

```
}
void loop()
{
    // The program block outputs an analog voltage value (0~1024). The value of this value is a proportional conversion of the soil moisture.
    Soil_moisture = analogRead(A3);
    if (50 > Soil_moisture) {
      // When the soil moisture is less than the set threshold, the sound and light alarm function is executed
      Watering_warning();
    }
    else
    {
      digitalWrite(13,LOW);
```

```
        noTone(3);


    }
}
```

Test Result：

When the soil moisture sensor detects that the soil moisture value is less than 50, the sound and light alarm is activated, the red LED light of the sound and light alarm will flash, and the buzzer will emit a ticking sound.

Note: The soil moisture sensor must be inserted into the soil of the plant. If it is not inserted into the soil, the humidity detected by the sensor is 0, which means extreme drought and the alarm will always sound.

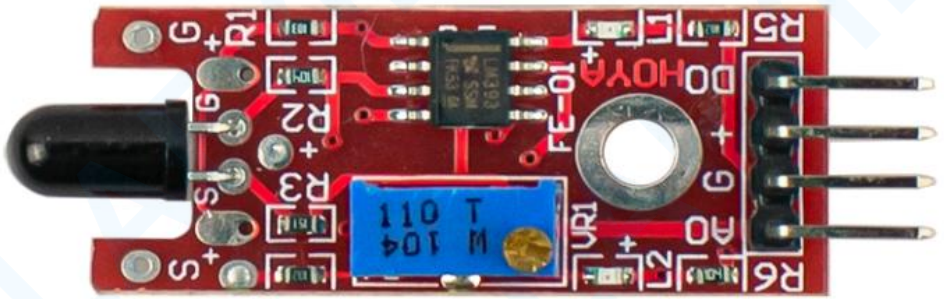
(Note: Before uploading the code, you need to disconnect the Bluetooth module from the sensor shield v5. After successfully uploading the code, reinstall the Bluetooth module.)

Project 5：Flame Alarm

Overview

Life is fragile and priceless, and only when we are alive can we pursue the next step in life. As the society reports more and more home life safety accidents, people are paying more attention to the safety of the living environment, and some alarms have been used more widely, especially for fire defense alarm systems. When the flame sensor detects that there is a flame, the fan is controlled to start, and the wind is used to blow the flame out.

Flame sensor



The flame sensor is a sensor device that uses infrared light to be sensitive to flames to detect flames. When In the event of a fire, there will be particularly strong infrared radiation. The flame sensor uses infrared rays to Sensitive features, using a special infrared receiver tube to detect the flame.It will react under the irradiation of infrared light of a specific band, and then The brightness of the flame is transformed into a level signal that changes from high to low. When in use,Keep a certain distance between the sensor and the flame to avoid high temperature damage to the sensor The test flame distance range of the lighter is 80cm, and the more the flame is Larger, the farther the detection distance.The flame sensor has four pins G, V, A, D, which can be connected to the main The digital interface or analog interface of the board is connected for use. If it is connected to the module The simulation value of the flame sensor will be based on the flame size and the distance from the flame.The distance becomes smaller as it gets closer and farther; if it is connected to a digital interface, the flame is transmitted.The digital value of the sensor is when the read analog value is less than the threshold, it outputs low level (0), if it is greater than the threshold, then Output high level (1). The size of the threshold can be adjusted by adjusting the size of the adjustable resistor.

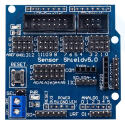
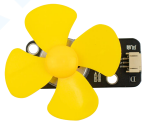
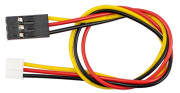
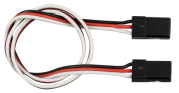
DC motor fan module

A DC motor is a motor that converts DC electrical energy into mechanical energy. Because of its good speed regulation performance, it is widely used in electric drive. According to the excitation mode, DC motors are divided into three types: permanent magnet, separate excitation and self-excitation. Among them, self-excitation is divided into three types: parallel excitation, series excitation and compound excitation. When the DC power supply supplies power to the armature winding through the brush, the current in the same direction can flow through the N-pole lower conductor on the armature surface. According to the left-hand rule, the conductor will be subjected to counterclockwise torque; the S-pole lower part of the armature surface The conductor also flows in the same direction, and according to the left-hand rule, the conductor will also be subjected to a counterclockwise moment. In this way, the entire armature winding, that is, the rotor, will rotate counterclockwise, and the input DC electrical energy will be converted into mechanical energy output on the rotor shaft. Composed of stator and rotor, stator: base, main magnetic pole, commutating pole, brush device, etc.; rotor

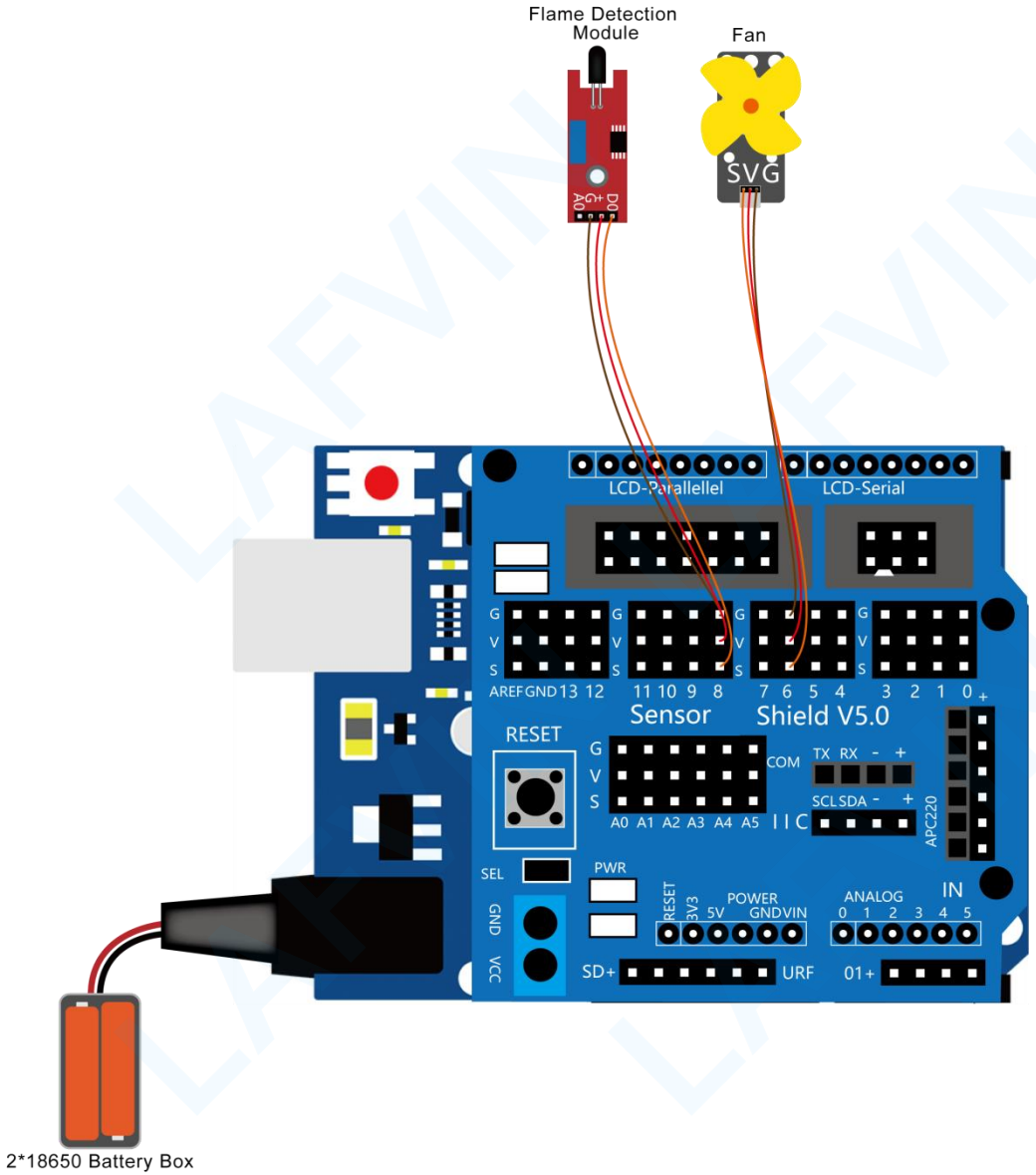
(Armature): Armature core, armature winding, commutator, shaft and fan, etc.

The motor fan module has three pin interfaces, G means GND grounding, V means VCC Connect to high level, S is the control signal input pin, which can be connected to the Arduino control board.Word port. The fan speed can also be adjusted by connecting the PWM signal pins of the digital port (D3 D5 D6 D9 D10 D11).

Experiment equipment:

Arduino UNO*1	Arduino Sensor Shield V5.0*1	USB cable*1	Flame sensor*1	Fan Module*1	3pin PH2.0 to Dupont line*1	3pin F-F Dupont line*1
						

Wiring Diagram：



Let's program

Project purpose

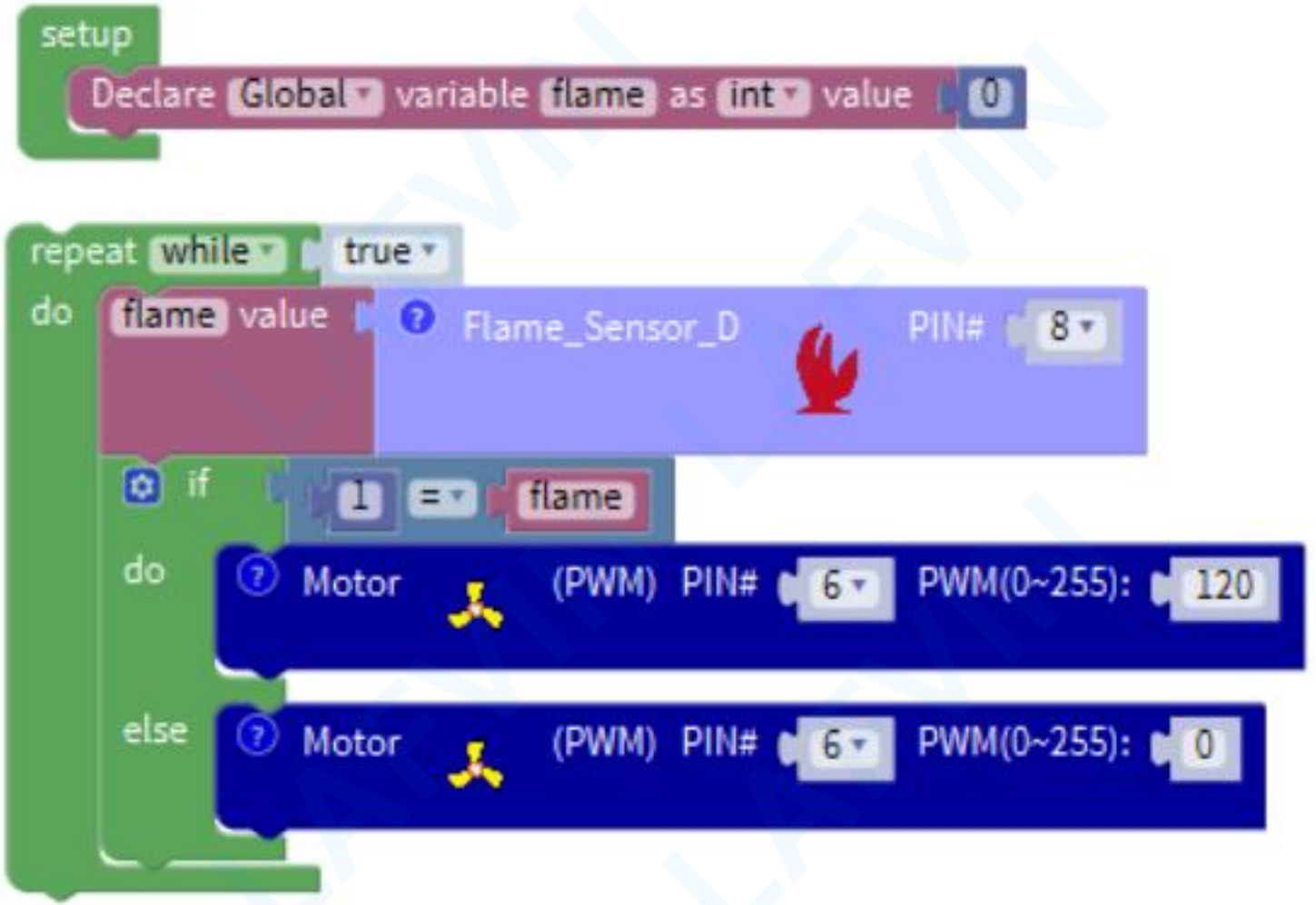
Write code to control the realization function:

When the flame sensor detects that there is a flame, the fan is controlled to start, and the wind is used to blow the flame out.

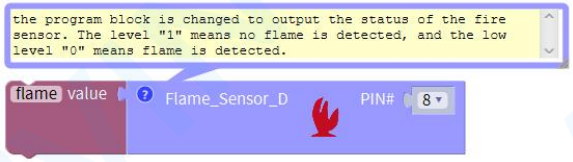
Mixly Code

Wire it up well as the above diagram. Okay, let’s move on to write the test code.Open Mixly software.Click “**New**” to add new project, then start your programming .

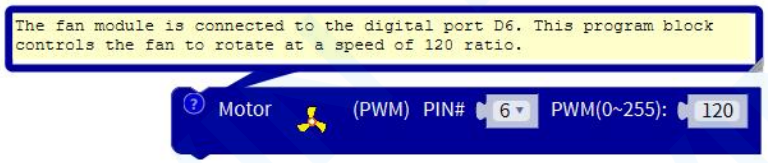
if you want to refer to the program we provide.open the reference code for this project"**Flame Alarm.mix**" in the reference materials we provided.


setup
Declare Global variable flame as int value 0
repeat while true
do
flame value
Flame_Sensor_D
PIN# 8
if 1 = flame
do
Motor (PWM) PIN# 6 PWM(0~255): 120
else
Motor (PWM) PIN# 6 PWM(0~255): 0

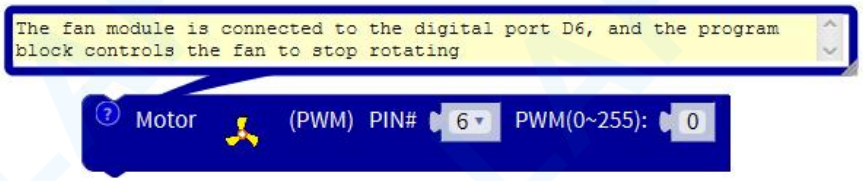
Programming Thinking



The flame sensor is connected to the digital port D8. This program block outputs the status of the fire sensor. A level "1" means no flame is detected, and a low level "0" means flame is detected. And save the obtained state value to the variable flame



The fan module is connected to the digital port D6. This program block controls the fan to rotate at a speed of 120 ratio.



The fan module is connected to the digital port D6, and the program block controls the fan to stop rotating

Arduino Code

If you want to refer to the program we provide.Open the reference code for this project "**Flame_Alarm.ino**" in the reference materials we provided.

Programming Thinking

```
/*
LAFVIN Smart Home Kit for Arduino
Project 5
*/
int flame;
//Define fan control function
void fan_motor_pwm(int speedpin, int speed)
{
```

```
    if (speed <= 0)
    {
        analogWrite(speedpin, 0);
    }
    else if (speed > 255)
    {
        analogWrite(speedpin, 255);
    }
    else
    {
        analogWrite(speedpin,speed);
    }
}
```

```
void setup(){
    flame = 0;
    pinMode(8, INPUT);
    pinMode(6, OUTPUT);
    digitalWrite(6, LOW);
}

void loop(){
        // the program block is changed to output the status of the fire sensor. The level "1" means no flame is detected, and the low level "0" means flame is detected.
        flame = digitalRead(8);
        if (1 == flame)
```

```
    {
        // The fan module is connected to the digital port D6. This program block controls the fan to rotate at a speed of
120 ratio.
        fan_motor_pwm(6, 120);

    }
    else
    {
        // The fan module is connected to the digital port D6, and the program block controls the fan to stop rotating
        fan_motor_pwm(6, 0);

    }
}
```

Test Result:

After uploading the code, use a lighter to ignite near the flame sensor, and the fan will start until the flame is blown out before stopping. If the presence of flame is not detected, the fan will not start.

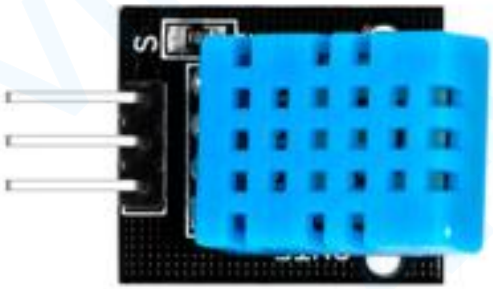
(Note: Before uploading the code, you need to disconnect the Bluetooth module from the sensor shield v5. After successfully uploading the code, reinstall the Bluetooth module.)

Project 6：Intelligent Temperature Control Fan

Overview

With the increasingly severe global warming trend, fans and air conditioners have become essential facilities for people's lives. However, air conditioners consume large amounts of energy and the incidence of air conditioning diseases is increasing year by year. Therefore, many people choose fans, which are low-energy-consumption cooling appliances. At present, the fans used in people's daily life use buttons or knobs to adjust the speed and wind, but is it possible to adjust the wind according to the temperature and humidity information of the indoor environment? If the fan can change the wind power according to the indoor temperature, it will undoubtedly be more beneficial to human health and save energy. In this lesson, we will make a fan with intelligent temperature control and see how it achieves the function of temperature control and wind.

Temperature and humidity sensor module



The temperature and humidity sensor module is a calibrated digital signal output Temperature and humidity composite sensor, it uses a dedicated digital module to collect Technology and temperature and humidity sensing technology to ensure that the product has extremely high reliability And excellent long-term stability.The temperature and humidity sensor module detects the surrounding environment through DHT11 Temperature and humidity, DHT11 includes a resistive humidity sensor And an NTC temperature measuring element, and a high-performance 8-bit microcontroller Connected, only one wire is needed to complete the data with Arduino transmission.

The temperature and humidity sensor module has three pins, G is GND grounded, V is VCC connected to high level or 5V, S Indicates the signal line, which can be connected to analog ports D0-D13 and A0-A5.

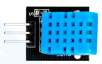
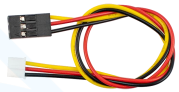
DC motor fan module

A DC motor is a motor that converts DC electrical energy into mechanical energy. Because of its good speed regulation performance, it is widely used in electric drive. According to the excitation mode, DC motors are divided into three types: permanent magnet, separate excitation and self-excitation. Among them, self-excitation is divided into three types: parallel excitation, series excitation and compound excitation. When the DC power supply supplies power to the armature winding through the brush, the current in the same direction can flow through the N-pole lower conductor on the armature surface. According to the left-hand rule, the conductor will be subjected to counterclockwise torque; the S-pole lower part of the armature surface The conductor also flows in the same direction, and according to the left-hand rule, the conductor will also be subjected to a counterclockwise moment. In this way, the entire armature winding, that is, the rotor, will rotate counterclockwise, and the input DC electrical energy will be converted into mechanical energy output on the rotor shaft. Composed of stator and rotor, stator: base, main magnetic pole, commutating pole, brush device, etc.; rotor

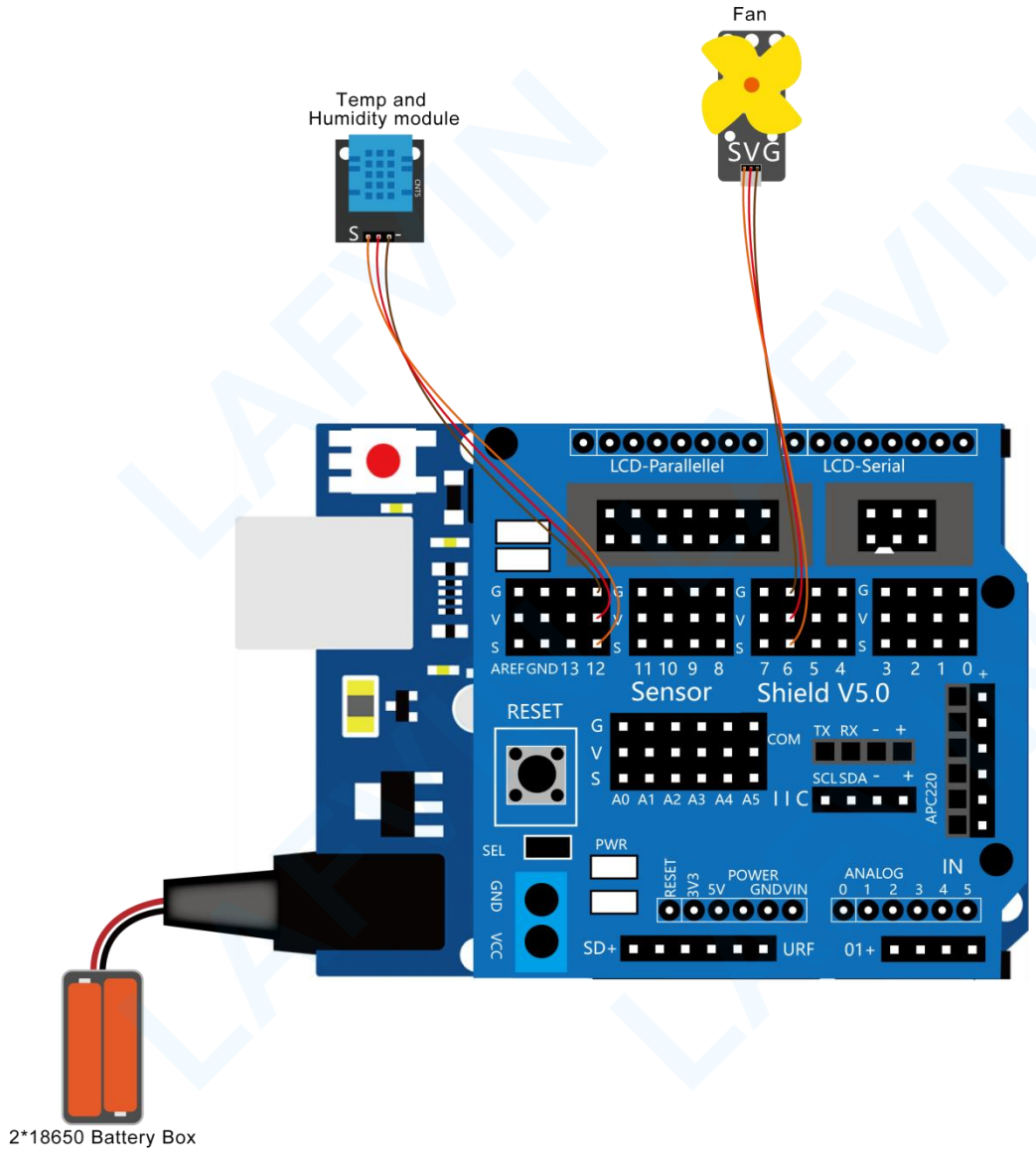
(Armature): Armature core, armature winding, commutator, shaft and fan, etc.

The motor fan module has three pin interfaces, G means GND grounding, V means VCC Connect to high level, S is the control signal input pin, which can be connected to the Arduino control board.Word port. The fan speed can also be adjusted by connecting the PWM signal pins of the digital port (D3 D5 D6 D9 D10 D11).

Experiment equipment:

Arduino UNO*1	Arduino Sensor Shield V5.0*1	USB cable*1	Temperature and humidity sensor module *1	Fan Module*1	3pin PH2.0 to Dupont line*1	3pin F-F Dupont line*1
						

Wiring Diagram：



Let's program

Project purpose

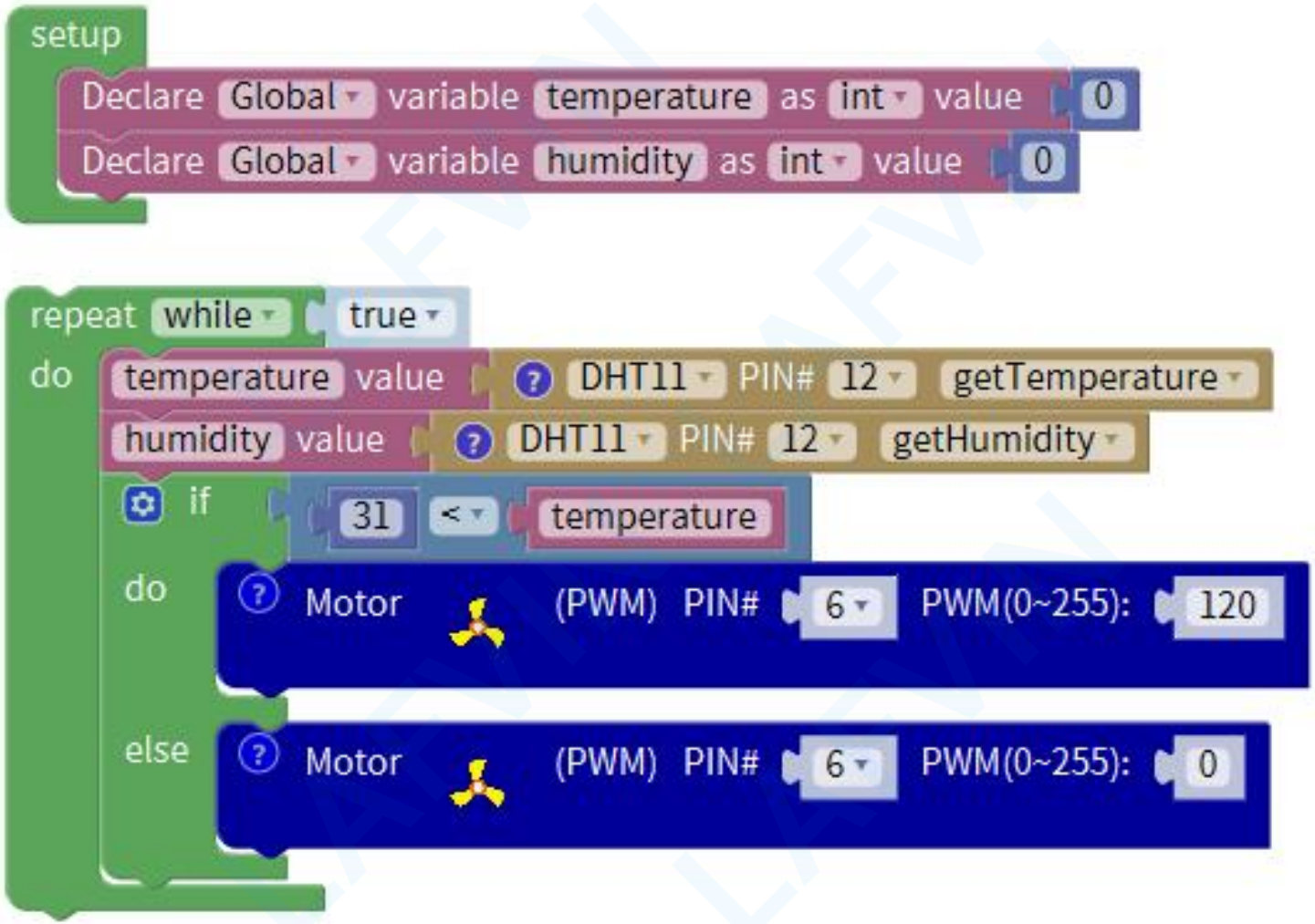
Write code to achieve the following functions:

When the program upload is completed, the temperature and humidity sensor is always in the detection state. When the ambient temperature is greater than 31 degrees Celsius, the fan starts at a proportional speed of 120 to dissipate the air in the home. The fan will not stop until the ambient temperature is below 31 degrees Celsius.

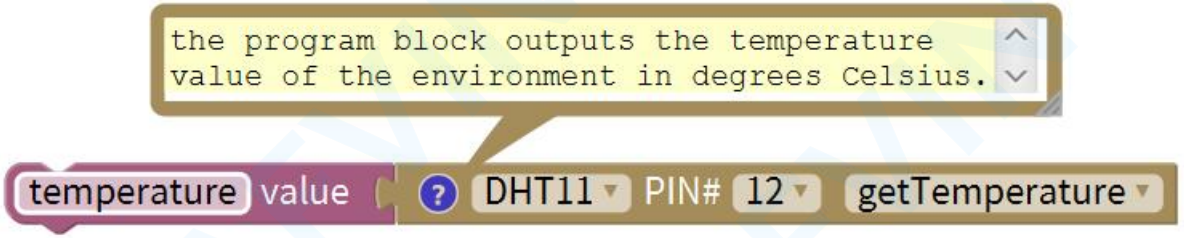
Mixly Code

Wire it up well as the above diagram. Okay, let’s move on to write the test code.Open Mixly software.Click “**New**” to add new project, then start your programming .

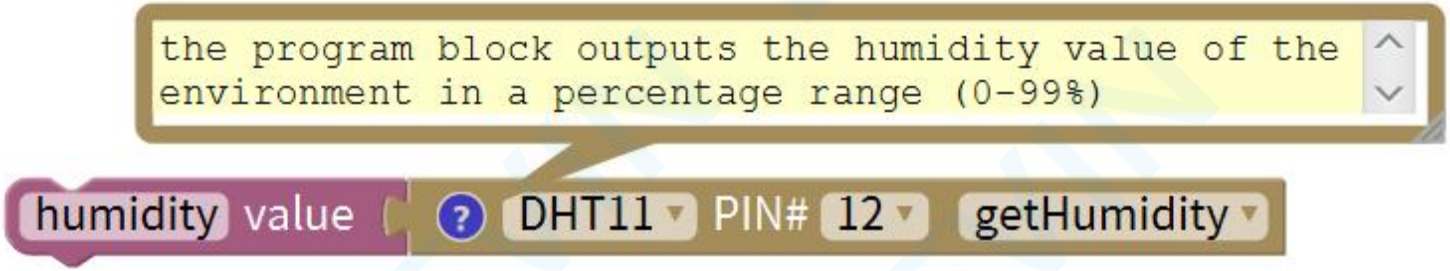
if you want to refer to the program we provide.open the reference code for this project"**Intelligent Temperature Control Fan.mix**" in the reference materials we provided.


setup
Declare Global variable temperature as int value 0
Declare Global variable humidity as int value 0
repeat while true
do
temperature value DHT11 PIN# 12 getTemperature
humidity value DHT11 PIN# 12 getHumidity
if 31 < temperature
do
Motor (PWM) PIN# 6 PWM(0~255): 120
else
Motor (PWM) PIN# 6 PWM(0~255): 0

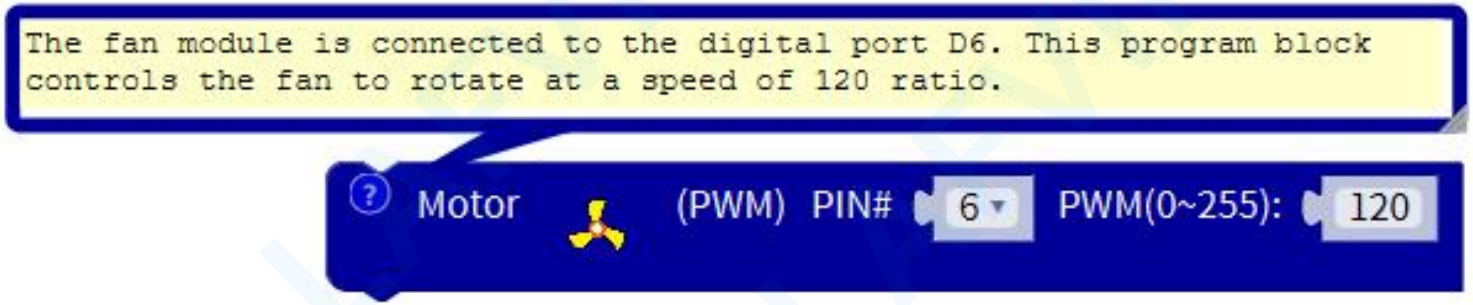
Programming Thinking



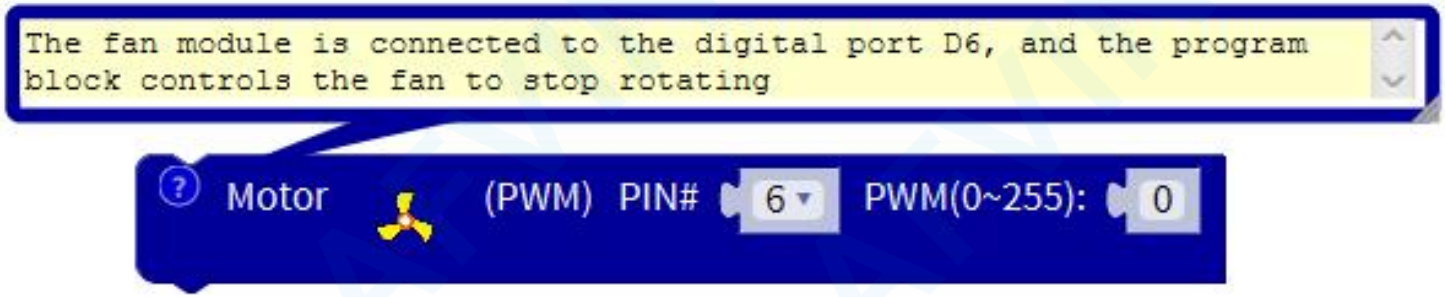
The temperature and humidity sensor is connected to the digital port D12, the program block outputs the temperature value of the environment in degrees Celsius. Save the temperature value to the variable temperature



The temperature and humidity sensor is connected to the digital port D12, and the program block outputs the humidity value of the environment in a percentage range (0-99%). Save the humidity value to the variable humidity



The fan module is connected to the digital port D6. This program block controls the fan to rotate at a speed of 120 ratio.



The fan module is connected to the digital port D6, and the program block controls the fan to stop rotating

Arduino Code

If you want to refer to the program we provide.Open the reference code for this project "**Intelligent_Temperature_Control_Fan.ino**" in the reference materials we provided.

Programming Thinking

```
/*
LAFVIN Smart Home Kit for Arduino
Project 6
*/
#include "DHT.h"
//Call the DHT.h library file
int temperature;
```

```
int humidity;
DHT dht12(12, 11);



//Define fan control function
void fan_motor_pwm(int speedpin, int speed)
{
    if (speed <= 0)
    {
        analogWrite(speedpin, 0);
    }
    else if (speed > 255)
    {
```

```
        analogWrite(speedpin, 255);
    }
    else
    {
        analogWrite(speedpin,speed);
    }
}

void setup(){
    temperature = 0;
    humidity = 0;
      dht12.begin();//Initialize the sensor
    pinMode(6, OUTPUT);
```

```
    digitalWrite(6, LOW);
}

void loop(){
    // the program block outputs the temperature value of the environment in degrees Celsius.
    temperature = dht12.readTemperature();
    // the program block outputs the humidity value of the environment in a percentage range (0-99%)
    humidity = dht12.readHumidity();
    if (31 < temperature) {
        // The fan module is connected to the digital port D6. This program block controls the fan to rotate at a speed of
120 ratio.
        fan_motor_pwm(6, 120);
    } else {
```

```
        // The fan module is connected to the digital port D6, and the program block controls the fan to stop rotating
        fan_motor_pwm(6, 0);
    }
}
```

Test Result:

After uploading the code, the temperature and humidity sensor is always in the detection state. When the ambient temperature is greater than 31 degrees Celsius, the fan starts and starts at a proportional speed of 120 to dissipate the air in the home. The fan will not stop until the ambient temperature is below 31 degrees Celsius.

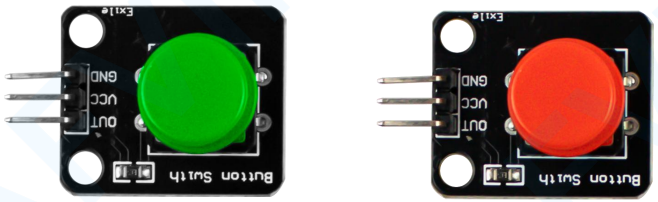
(Note: Before uploading the code, you need to disconnect the Bluetooth module from the sensor shield v5. After successfully uploading the code, reinstall the Bluetooth module.)

Project 7：Password Door

Overview

In order to protect the safety of their homes and private property, people invented door locks. In ancient times, people locked their doors with ropes or wooden bolts, and then developed to use key door locks. People often left the keys at home or lost them, causing the distress of not being able to open the door, and its safety is not Very high. Therefore, more intelligent, safe and convenient door locks such as fingerprint door locks and password door locks have emerged. The task of this lesson is to make a smart password automatic door, let us understand the principle and operation of this access control operation.

Button Module



I believe that button switch is well-known by people. It belongs to switch quantity(digital quantity)component. Composed of normally open contact and normally closed contact,its working principle is similar with ordinary switch. When the normally open contact bears pressure, the circuit is on state ; however,when this pressure disappears, the normally open contact goes back to initial state, that is, off state. The pressure is the act we switch the button.

Two button modules are used in use, namely the red and green button modules. The red button is connected to the digital port D5, the green button is connected to the digital port D4, the red button is used as the password input button, and the

green button is used as the password confirmation button.

When the red button is pressed, the signal output of the button is low level "0", when the button is released, the signal output is high level "1". Imitate the principle of the telegraph during the war. In this project, we cleverly used the length of time pressed to distinguish different key values.

When the red button is pressed for more than 500 milliseconds, the entered password value is "-"

When the red button is pressed for less than 500 milliseconds, the entered password value is "."

In the experimental reference program, we set the password as ".--.". When the password is entered, press the green button to confirm the password. If the entered password is equal to ".--.", the password is correct, and the LCD displays Password: Right. If the password is not equal to ".--.", the password is wrong, and the LCD displays Password: Error. Then wait for one second and the LCD displays Password: Again. At this time, press the red button again to enter the password.

LCD 1602 Display

With I2C communication module, this is a display module that can show 2 lines with 16 characters per line.It shows blue background and white word and connects to I2C interface of MCU,which highly save the MCU resources.On the back of LCD display, there is a blue potentiometer for adjusting the backlight. The communication address defaults to 0x27.The original 1602 LCD can start and run with 7 IO ports, but ours is built with Arduino IIC/I2C interface, saving 5 IO ports. Alternatively, the module comes with 4 positioning holes with a diameter of 3mm, which is convenient for you to fix on other devices.

Specifications:

I2C address: 0x27

Backlight (blue, white)

Power supply voltage: 5V

Adjustable contrast

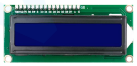
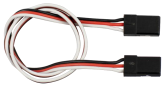
GND: A pin that connects to ground

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

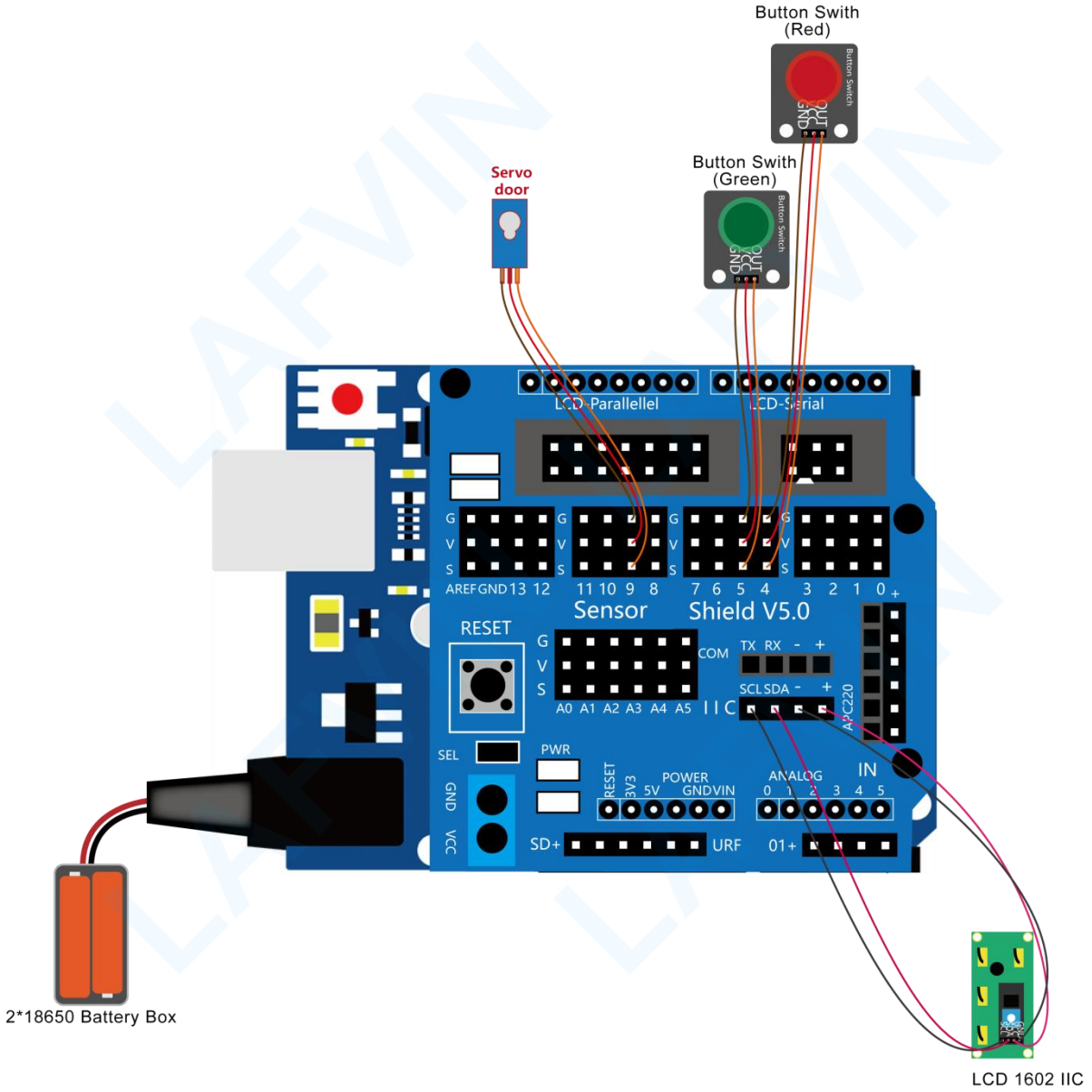
SDA: A pin that connects to analog port A4 for IIC communication

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

Experiment equipment:

Arduino UNO*1	Arduino Sensor Shield V5.0*1	USB cable*1	Button Module(red and white)*2	Servo*1	1602 LCD Display*1	2pin F-F Dupont line*2	3pin F-F Dupont line*1
							

Wiring Diagram：



Let's program

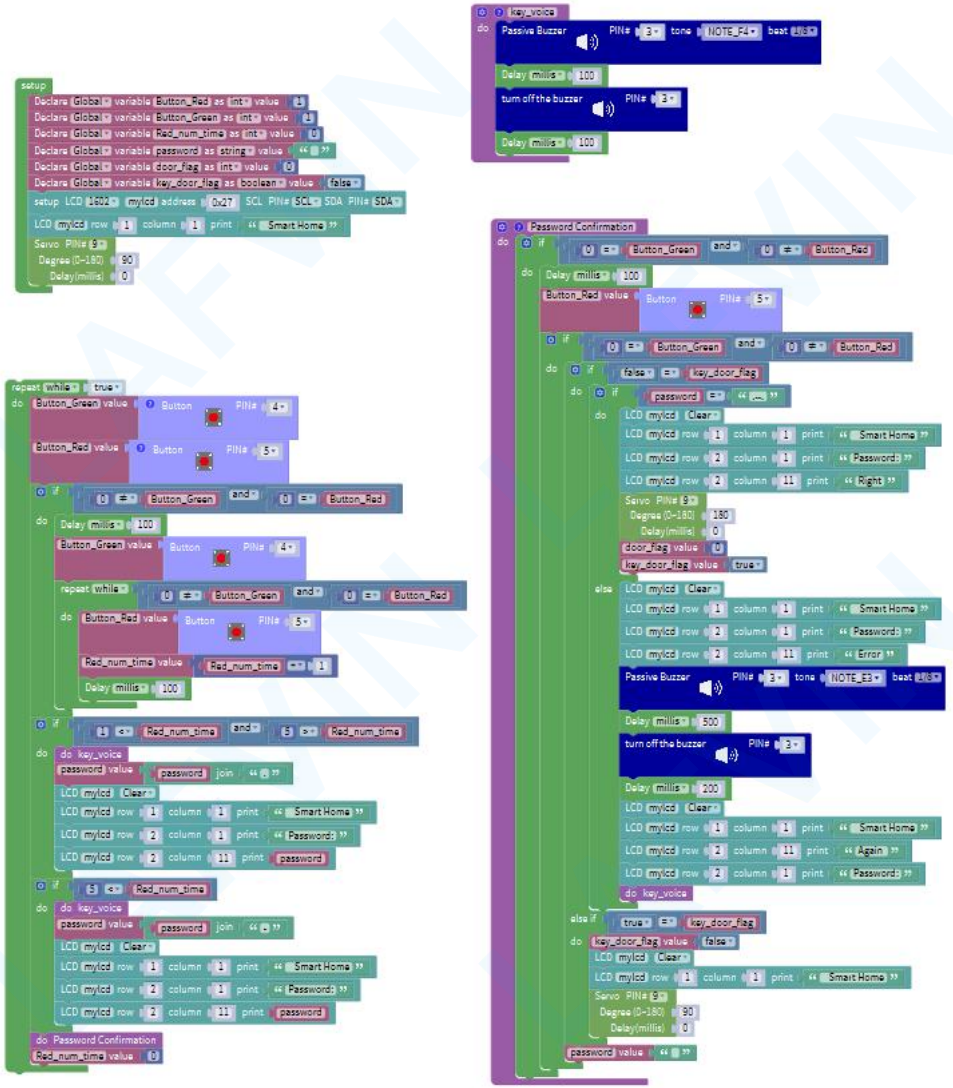
Project purpose

In the experimental reference program, we set the password as ".--.". When the password is entered, press the green button to confirm the password. If the entered password is equal to ".--.", the password is correct, and the LCD displays Password: Right. If the password is not equal to ".--.", the password is wrong, and the LCD displays Password: Error. Then wait for one second and the LCD displays Password: Again. At this time, press the red button again to enter the password.

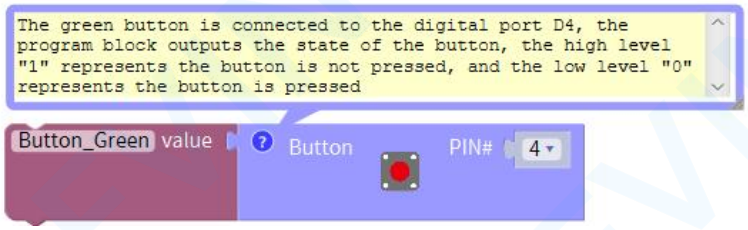
Mixly Code

Wire it up well as the above diagram. Okay, let’s move on to write the test code.Open Mixly software.Click “**New**” to add new project, then start your programming .

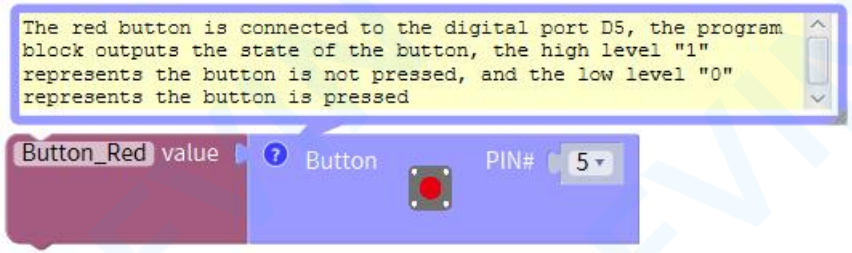
if you want to refer to the program we provide.open the reference code for this project"**Password Door.mix**" in the reference materials we provided.


setup
Declare Global variable Button_Red as int value 0
Declare Global variable Button_Green as int value 0
Declare Global variable Red_num_time as int value 0
Declare Global variable password as string value ""
Declare Global variable door_flag as int value 0
Declare Global variable key_door_flag as boolean value false
setup LCD 1602 mylcd address 0x27 SCL PIN# SCL SDA PIN# SDA
LCD mylcd row 1 column 1 print "Smart Home"
Servo PIN# 5
Degree (0~180) 90
Delay(millis) 0
repeat while true
do
Button_Green value Button PIN# 4
Button_Red value Button PIN# 5
if 0 = Button_Green and 0 = Button_Red
do
Delay millis 100
Button_Green value Button PIN# 4
repeat while 0 = Button_Green and 0 = Button_Red
do
Button_Red value Button PIN# 5
Red_num_time value Red_num_time - 1
Delay millis 100
if 1 <= Red_num_time and 5 >= Red_num_time
do
do key_voice
password value password join "0"
LCD mylcd Clear
LCD mylcd row 1 column 1 print "Smart Home"
LCD mylcd row 2 column 1 print "Password:"
LCD mylcd row 2 column 11 print password
if 5 <= Red_num_time
do
do key_voice
password value password join "0"
LCD mylcd Clear
LCD mylcd row 1 column 1 print "Smart Home"
LCD mylcd row 2 column 1 print "Password:"
LCD mylcd row 2 column 11 print password
do Password Confirmation
Red_num_time value 0
key_voice
do
Passive Buzzer PIN# 3 tone NOTE_F4 beat 1/8
Delay millis 100
turn off the buzzer PIN# 3
Delay millis 100
Password Confirmation
do
if 0 = Button_Green and 0 = Button_Red
do
Delay millis 100
Button_Red value Button PIN# 5
if 0 = Button_Green and 0 = Button_Red
do
if false = key_door_flag
do
if password = "0"
do
LCD mylcd Clear
LCD mylcd row 1 column 1 print "Smart Home"
LCD mylcd row 2 column 1 print "Password:"
LCD mylcd row 2 column 11 print "Right"
Servo PIN# 5
Degree (0~180) 180
Delay(millis) 0
door_flag value 0
key_door_flag value true
else
LCD mylcd Clear
LCD mylcd row 1 column 1 print "Smart Home"
LCD mylcd row 2 column 1 print "Password:"
LCD mylcd row 2 column 11 print "Error"
Passive Buzzer PIN# 3 tone NOTE_E3 beat 1/8
Delay millis 500
turn off the buzzer PIN# 3
Delay millis 200
LCD mylcd Clear
LCD mylcd row 1 column 1 print "Smart Home"
LCD mylcd row 2 column 11 print "Again"
LCD mylcd row 2 column 1 print "Password:"
do key_voice
else if true = key_door_flag
do
key_door_flag value false
LCD mylcd Clear
LCD mylcd row 1 column 1 print "Smart Home"
Servo PIN# 5
Degree (0~180) 90
Delay(millis) 0
password value ""

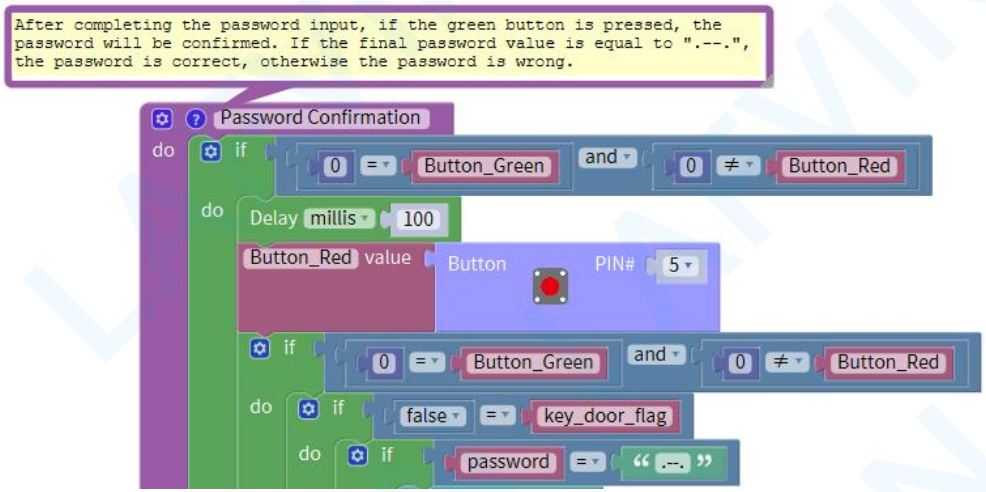
Programming Thinking



The green button is connected to the digital port D4, and the program block outputs the state of the button. The high level "1" represents the button is not pressed, and the low level "0" represents the button is pressed. Save the obtained button state value to the variable Button_Green



The red button is connected to the digital port D5. The program block outputs the state of the button. The high level "1" represents the button is not pressed, and the low level "0" represents the button is pressed. Save the obtained button state value to the variable Button_Red.



In the experimental reference program, we set the password as ".--.". When the password is entered, press the green button to confirm the password. If the entered password is equal to ".--.", the password is correct, and the LCD displays Password: Right. If the password is not equal to ".--.", the password is wrong, and the LCD displays Password: Error. Then wait for one second and the LCD displays Password: Again. At this time, press the red button again to enter the password.

Arduino Code

If you want to refer to the program we provide.Open the reference code for this project "**Password Door.ino**" in the reference materials we provided.

Before you can run this, make sure that you have installed the < Servo> <LiquidCrystal_I2C>library or re-install it, if necessary. Otherwise, your code won't work.For details about loading the library file, see Lesson**"2.3 How to Add Libraries"** about how to add libraries .

Programming Thinking

```
/*
LAFVIN Smart Home Kit for Arduino
Project 7
*/
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>

#include <Servo.h>

int Button_Red;

int Button_Green;

int Red_num_time;

String password;

int door_flag;

boolean key_door_flag;

LiquidCrystal_I2C mylcd(0x27,16,2);

Servo servo_9;


// Define the sound of password keys

void key_voice() {
```

```
    tone(3,349);
    delay(125);
    delay(100);
    noTone(3);
    delay(100);
}

// After completing the password input, if the green button is pressed, the password will be confirmed. If the final password value is equal to ".--.", the password is correct, otherwise the password is wrong.
void Password_Confirmation() {
    if (0 == Button_Green && 0 != Button_Red) {
        delay(100);
        Button_Red = digitalRead(5);
```

```
if (0 == Button_Green && 0 != Button_Red) {
    if (false == key_door_flag) {
        if (password == ".--.") {
            mylcd.clear();
            mylcd.setCursor(1-1, 1-1);
            mylcd.print("      Smart Home");
            mylcd.setCursor(1-1, 2-1);
            mylcd.print("Password:");
            mylcd.setCursor(11-1, 2-1);
            mylcd.print("Right");
            servo_9.write(180);
            delay(0);
            door_flag = 0;
```

```
    key_door_flag = true;


} else {
    mylcd.clear();
    mylcd.setCursor(1-1, 1-1);
    mylcd.print("      Smart Home");
    mylcd.setCursor(1-1, 2-1);
    mylcd.print("Password:");
    mylcd.setCursor(11-1, 2-1);
    mylcd.print("Error");
    tone(3,165);
    delay(125);
    delay(500);
```

```
    noTone(3);
    delay(200);
    mylcd.clear();
    mylcd.setCursor(1-1, 1-1);
    mylcd.print("      Smart Home");
    mylcd.setCursor(11-1, 2-1);
    mylcd.print("Again ");
    mylcd.setCursor(1-1, 2-1);
    mylcd.print("Password:");
    key_voice();

}
```

```
        } else if (true == key_door_flag) {
            key_door_flag = false;
            mylcd.clear();
            mylcd.setCursor(1-1, 1-1);
            mylcd.print("      Smart Home");
            servo_9.write(90);
            delay(0);
        }
        password = "";

    }

}
```

```
}


void setup(){
    pinMode(3, OUTPUT);
    Button_Red = 1;
    Button_Green = 1;
    Red_num_time = 0;
    password = "";
    door_flag = 0;
    key_door_flag = false;
    mylcd.init();
    mylcd.backlight();
    servo_9.attach(9);
```

```
    mylcd.setCursor(1-1, 1-1);
    mylcd.print("      Smart Home");
    servo_9.write(90);
    delay(0);
    pinMode(5, INPUT);
    pinMode(4, INPUT);
}

void loop(){
      // The green button is connected to the digital port D4, the program block outputs the state of the button, the high level "1" represents the button is not pressed, and the low level "0" represents the button is pressed
      Button_Green = digitalRead(4);
      // The red button is connected to the digital port D5, the program block outputs the state of the button, the high level
```

"1" represents the button is not pressed, and the low level "0" represents the button is pressed

```
    Button_Red = digitalRead(5);
    if (0 != Button_Green && 0 == Button_Red) {
      delay(100);
      Button_Green = digitalRead(4);
      while (0 != Button_Green && 0 == Button_Red) {
        Button_Red = digitalRead(5);
        Red_num_time = Red_num_time + 1;
        delay(100);
      }

    }
    if (1 < Red_num_time && 5 > Red_num_time) {
```

```
    key_voice();
    password = String(password) + String(".");
    mylcd.clear();
    mylcd.setCursor(1-1, 1-1);
    mylcd.print("      Smart Home");
    mylcd.setCursor(1-1, 2-1);
    mylcd.print("Password:");
    mylcd.setCursor(11-1, 2-1);
    mylcd.print(password);


}
if (5 < Red_num_time) {
    key_voice();
```

```
        password = String(password) + String("-");
        mylcd.clear();
        mylcd.setCursor(1-1, 1-1);
        mylcd.print("      Smart Home");
        mylcd.setCursor(1-1, 2-1);
        mylcd.print("Password:");
        mylcd.setCursor(11-1, 2-1);
        mylcd.print(password);

    }
    Password_Confirmation();
    Red_num_time = 0;
}
```

Test Result:

When the red button is pressed for more than 500 milliseconds, the entered password value is "-"

When the red button is pressed for less than 500 milliseconds, the entered password value is "."

In the experimental reference program, we set the password as ".--.".

When the password is entered, press the green button to confirm the password. If the entered password is equal to ".--.", the password is correct, and the LCD displays Password: Right. If the password is not equal to ".--.", the password is wrong, and the LCD displays Password: Error. Then wait for one second and the LCD displays Password: Again. At this time, press the red button again to enter the password.

When the password is seriously correct, the door will open. If the green button is pressed again, the door can be closed manually.

(Note: Before uploading the code, you need to disconnect the Bluetooth module from the sensor shield v5. After successfully uploading the code, reinstall the Bluetooth module.)

Project 8：Multi-purpose Smart Home

Overview

In this project, we will perform all the functions together,

We will achieve the following effects:

(1) Sound Photosensitive Control LED.

Contains light sensor, sound sensor module and LED. When

At night, when someone passes by and makes the sound of footsteps, the LED lights up; no one is nearby,

The indicator light goes out.

(2) Coin Parking

Contains Photoelectric interrupt Module, Servo.

When a coin is inserted, the parking gate will open.

when the coin is taken out, the parking gate will be closed.

(3) Rain-Controlled Window

Contains Water Lever Detection Sensor Module, Servo.

the initial state of the window is open. The water level detection sensor is installed on the top of the house. When it rains, the sensor can directly detect the change of rainwater.

When it is raining, the water level detection sensor detects that the corresponding analog voltage value of the rainfall is greater than 100 and the window will be closed.

Otherwise the window is open

(4) Plant Watering Warning

Including Soil Humidity Sensor, Passive Buzzer, LED Module.

When the soil moisture sensor detects that the soil moisture value is less than 50, the sound and light alarm is activated, the red LED light of the sound and light alarm will flash, and the buzzer will emit a ticking sound.

Note: The soil moisture sensor must be inserted into the soil of the plant. If it is not inserted into the soil, the humidity detected by the sensor is 0, which means extreme drought and the alarm will always sound.

(5) Flame Alarm

Including Flame sensor, DC motor fan module.

After uploading the code, use a lighter to ignite near the flame sensor, and the fan will start until the flame is blown out before stopping. If the presence of flame is not detected, the fan will not start.

(6) Intelligent Temperature Control Fan

Including Temperature and humidity sensor module, DC motor fan module.

After uploading the code, the temperature and humidity sensor is always in the detection state. When the ambient temperature is greater than 31 degrees Celsius, the fan starts and starts at a proportional speed of 120 to dissipate the air in the home. The fan will not stop until the ambient temperature is below 31 degrees Celsius.

(7) Password Door

Including Button Module, 1602 LCD Display, Servo.

When the red button is pressed for more than 500 milliseconds, the entered password value is "-"

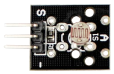
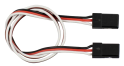
When the red button is pressed for less than 500 milliseconds, the entered password value is "."

In the experimental reference program, we set the password as ".--.".

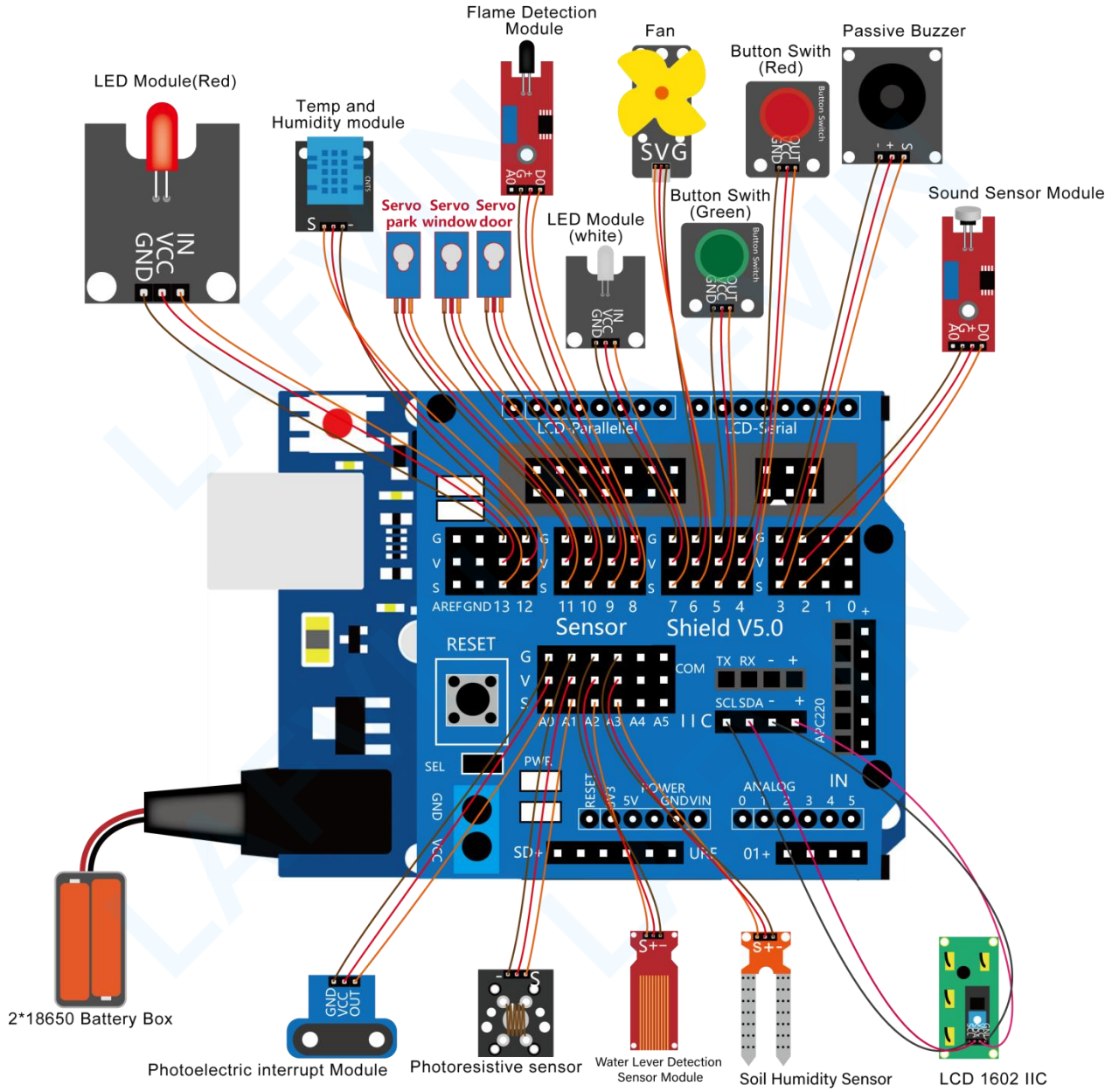
When the password is entered, press the green button to confirm the password. If the entered password is equal to ".--.", the password is correct, and the LCD displays Password: Right. If the password is not equal to ".--.", the password is wrong, and the LCD displays Password: Error. Then wait for one second and the LCD displays Password : Again. At this time, press the red button again to enter the password.

When the password is seriously correct, the door will open. If the green button is pressed again, the door can be closed manually.

Experiment equipment:

Arduino UNO*1	Arduino Sensor Shield V5.0*1	USB cable*1	Button Module*2	Servo*3	1602 LCD Display*1	Flame sensor*1	Fan Module*1
							
Photocell Sensor*1	Sound sensor module*1	LED module*2	Photoelectric Speed Sensor*1	Water Lever Detection Sensor*1	Soil Humidity Sensor*1	Passive Buzzer*1	Temperature and humidity sensor *1
							
2pin F-F Dupont line*2	3pin F-F Dupont line*12	3pin PH2.0 to Dupont line*1	Battery box*1				
							

Wiring Diagram：



Let's program

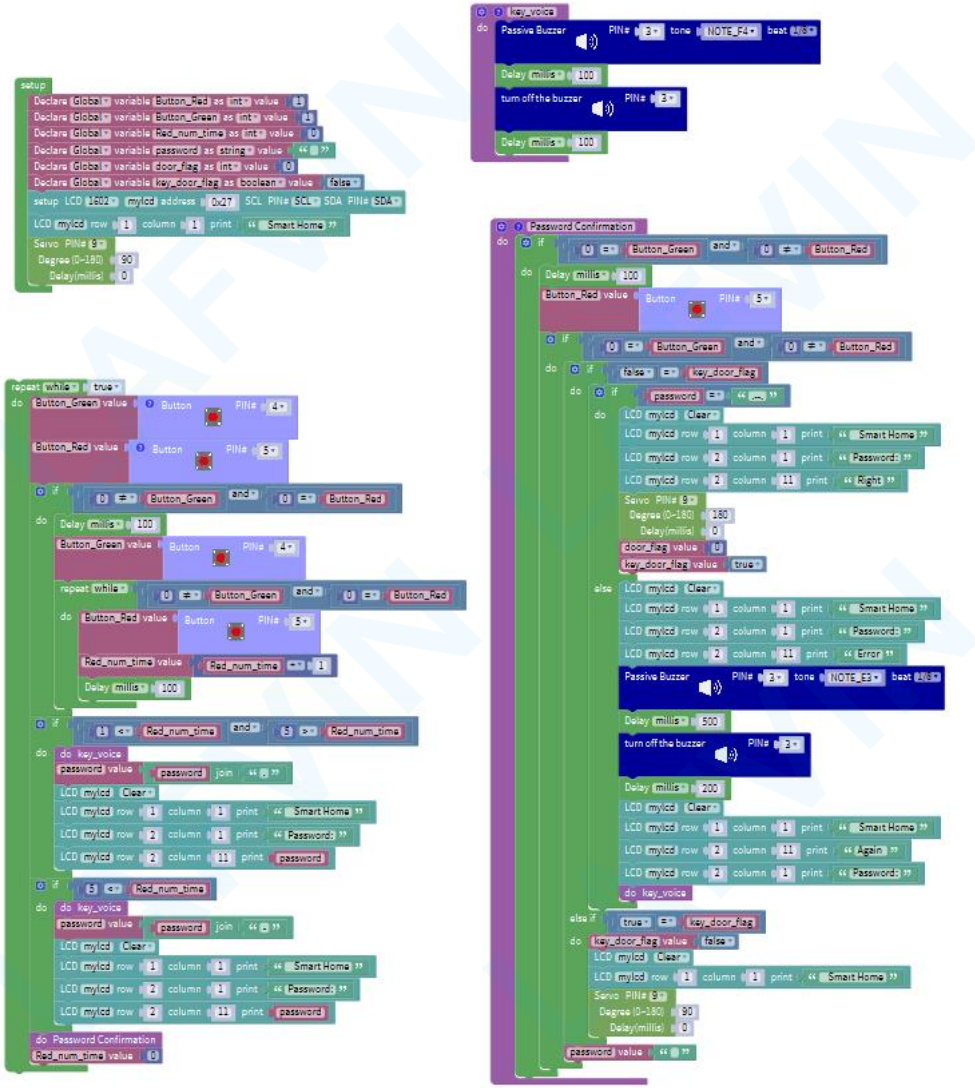
Project purpose

In the experimental reference program, we set the password as ".--.". When the password is entered, press the green button to confirm the password. If the entered password is equal to ".--.", the password is correct, and the LCD displays Password: Right. If the password is not equal to ".--.", the password is wrong, and the LCD displays Password: Error. Then wait for one second and the LCD displays Password: Again. At this time, press the red button again to enter the password.

Mixly Code

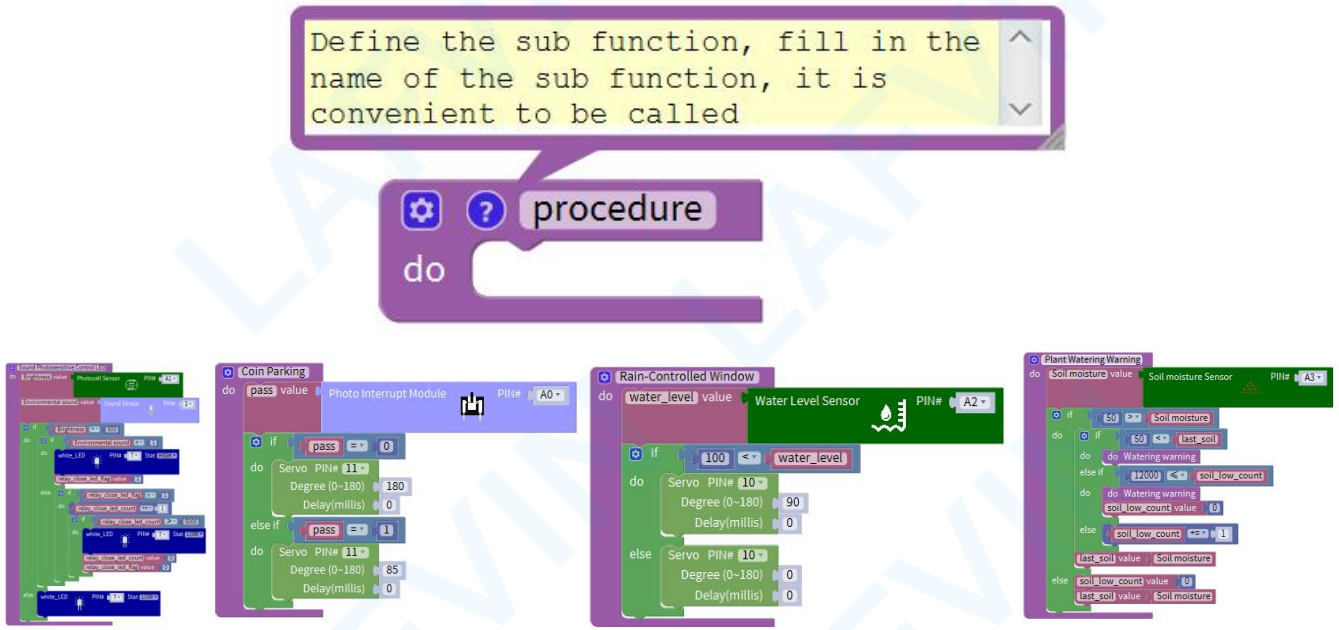
Wire it up well as the above diagram. Okay, let’s move on to write the test code.Open Mixly software.Click “**New**” to add new project, then start your programming .

if you want to refer to the program we provide.open the reference code for this project"**Multi-purpose Smart Home.mix**" in the reference materials we provided.

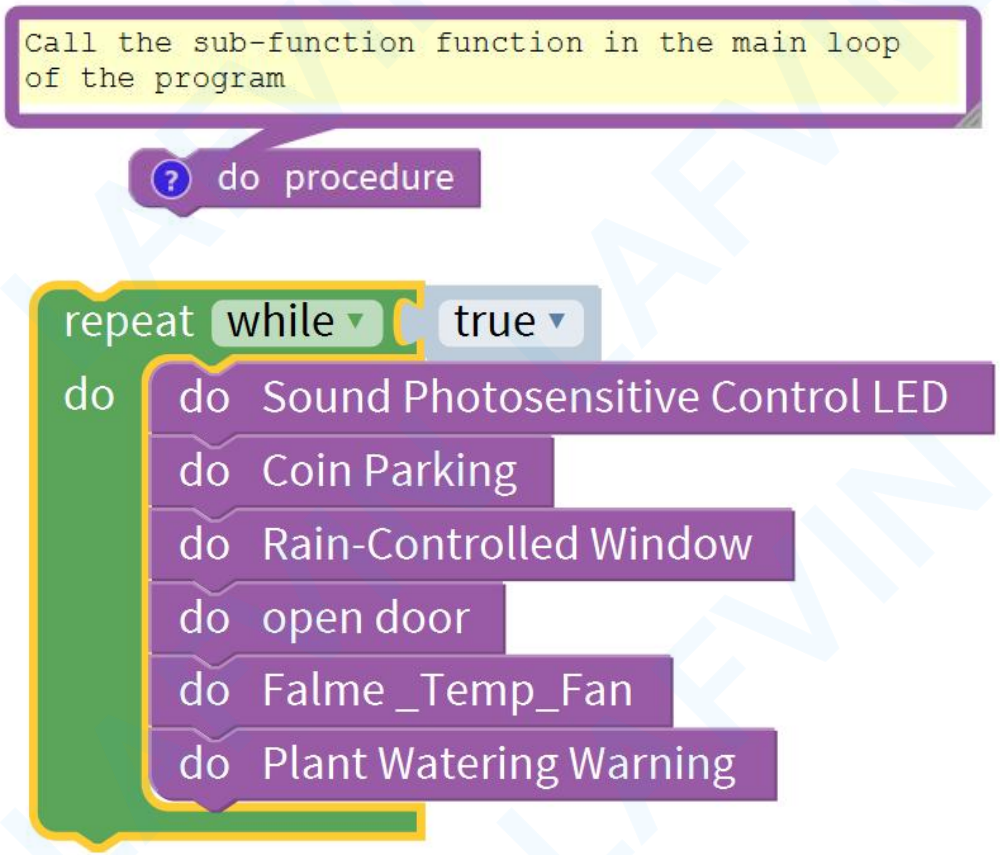

setup
Declare (Global) variable Button_Red as (int) value (4)
Declare (Global) variable Button_Green as (int) value (3)
Declare (Global) variable Red_num_time as (int) value (0)
Declare (Global) variable password as (string) value ""
Declare (Global) variable door_flag as (int) value (0)
Declare (Global) variable key_door_flag as (boolean) value (false)
setup LCD 1602 (mylcd) address 0x27 SCL PIN# SCL SDA PIN# SDA
LCD (mylcd) row 1 column 1 print "Smart Home"
Servo PIN# 9
Degree (0~180) 90
Delay(millis) 0
key_voice
Passive Buzzer PIN# 3 tone NOTE_F4 beat 1/8
Delay (millis) 100
turn off the buzzer PIN# 3
Delay (millis) 100
repeat while true
Button_Green value Button PIN# 4
Button_Red value Button PIN# 5
Delay (millis) 100
Red_num_time value Red_num_time + 1
password value password join "0"
Password Confirmation
Red_num_time value 0
Password Confirmation
Button_Green and Button_Red
Button_Red value Button PIN# 5
key_door_flag
password
Clear
Smart Home
Password:
Right
Error
Again
Passive Buzzer PIN# 3 tone NOTE_E3 beat 1/8
turn off the buzzer PIN# 3
Delay (millis) 500
Delay (millis) 200
door_flag value 0
key_door_flag value true
password value ""

Programming Thinking

Define the sub function, fill in the name of the sub function, it is convenient to be called



Call the sub-function function in the main loop of the program



Arduino Code

If you want to refer to the program we provide.Open the reference code for this project "**Multi-purpose Smart Home.ino**" in the reference materials we provided.

Before you can run this, make sure that you have installed the < Servo> <LiquidCrystal_I2C>library or re-install it, if necessary. Otherwise, your code won't work.For details about loading the library file, see Lesson**"2.3 How to Add Libraries"** about how to add libraries .

Test Result:

After uploading the code, all the functions of the previous courses are integrated. The functions can perform operations at the same time and provide comprehensive monitoring for the home. For example, the water level sensor module detects rain and controls the steering gear to close the windows. At the same time, if you want to park your car in the garage, the parking gate will open at the same time as long as you put a coin.(Note: Before uploading the code, you need to disconnect the Bluetooth module from the sensor shield v5. After successfully uploading the code, reinstall the Bluetooth module.)

Project 9：Bluetooth Test

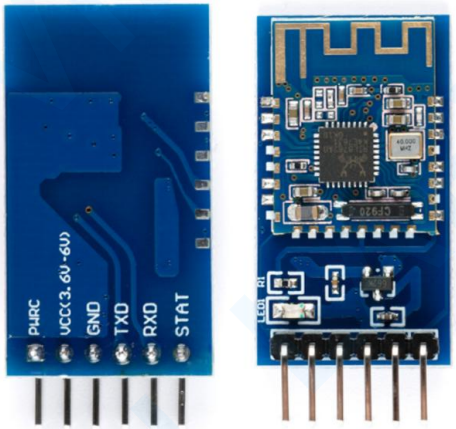
Overview

In the previous project course, we learned to execute all functions at the same time, but this is only through offline control. Will adding wireless control in a smart home make it smarter? In this project, learn the use of the Bluetooth module, and communicate with the Bluetooth module through the APP to achieve wireless control functions, such as controlling the turning on and off of the LED light module in the home through the APP, and controlling the rotation and stop of the fan.

Bluetooth Module JDY-16

The JDY-16 transparent transmission module is based on the Bluetooth-compatible 4.2 protocol standard, the working frequency band is 2.4GHZ range, the modulation method is GFSK, the maximum transmission power is 0db, the maximum transmission distance is 60 meters, and it adopts imported original chip design and supports users to modify the device through AT commands Name, service UUID, transmit power, pairing password and other instructions are convenient and flexible to use. The JDY-16 Bluetooth-compatible module can realize the data transmission between the module and the mobile phone or between the module and the module. The communication mode of UART or IIC can be selected through

IO, and the Bluetooth-compatible can be quickly used for product application through simple configuration. Make BLE's application in products faster and more convenient.



Product Parameters:

Model: JDY-16

Working frequency: 2.4G

Transmitting power: 0db (maximum)

Communication interface: UART or IIC

Working voltage: 1.8V-3.6V

Working temperature: -40℃-80℃

Antenna: Built-in PCB antenna

Receiving sensitivity: -97dbm

Transmission distance: 60 meters

Module size: 19.6mm * 14.94 *2.6

Bluetooth-compatible version: BLE 4.2 (compatible with BLE4.0, BLE4.1)

Transparent transmission rate: 115200 bps/s

Wake-up state current: 4mA (with broadcast)

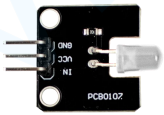
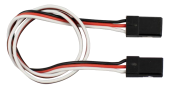
Light sleep state current: <300uA (with broadcast)

Deep sleep current: 1.8uA (no broadcast)

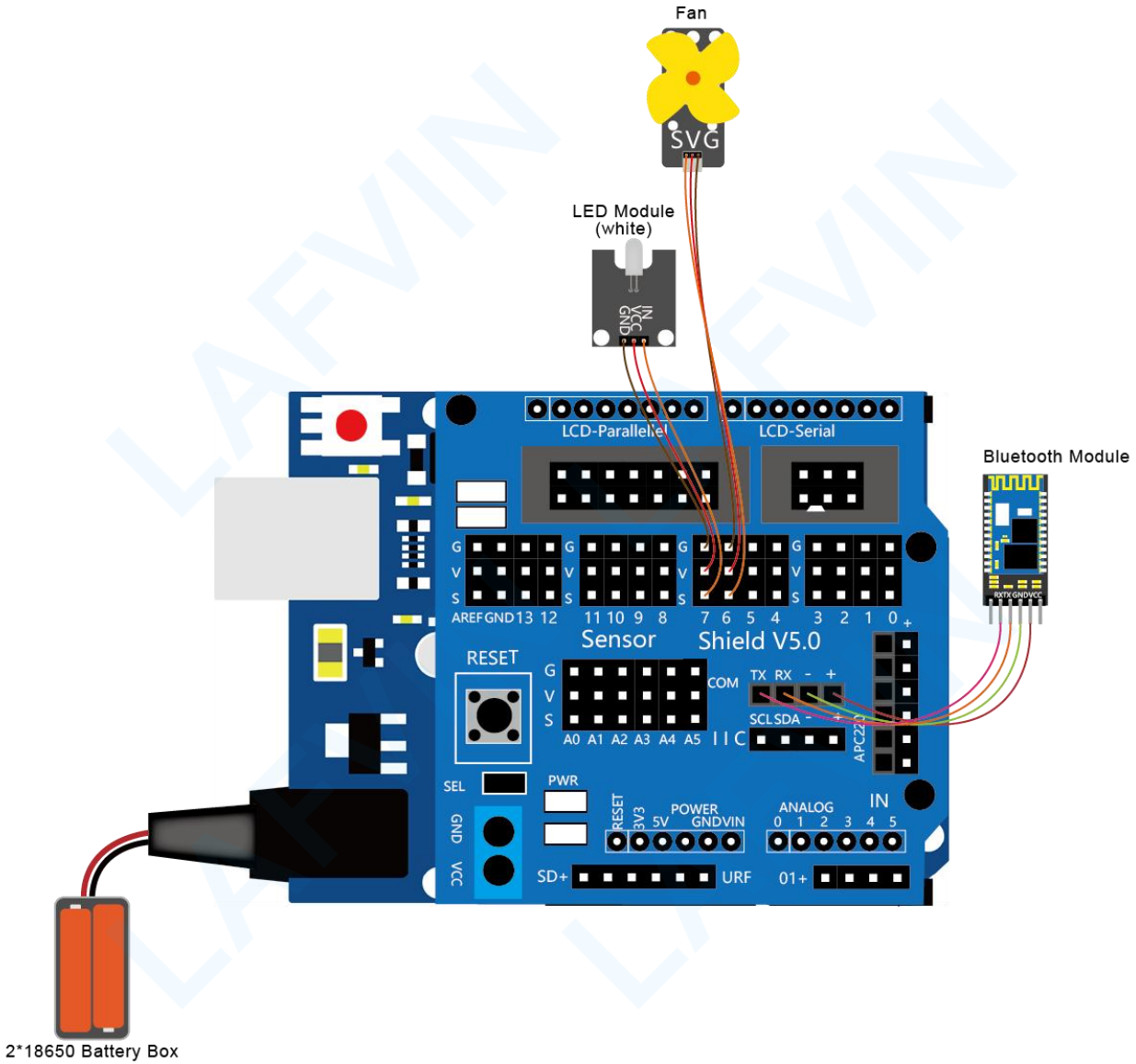
Command parameter saving: parameter configuration data is saved after power-off

STM welding temperature: <300℃

Experiment equipment:

Arduino UNO*1	Arduino Sensor Shield V5.0*1	USB cable*1	Bluetooth Module*1	Fan Module*1	LED Module*1	3pin PH2.0 to Dupont line*1	3pin F-F Dupont line*1
							

Wiring Diagram：

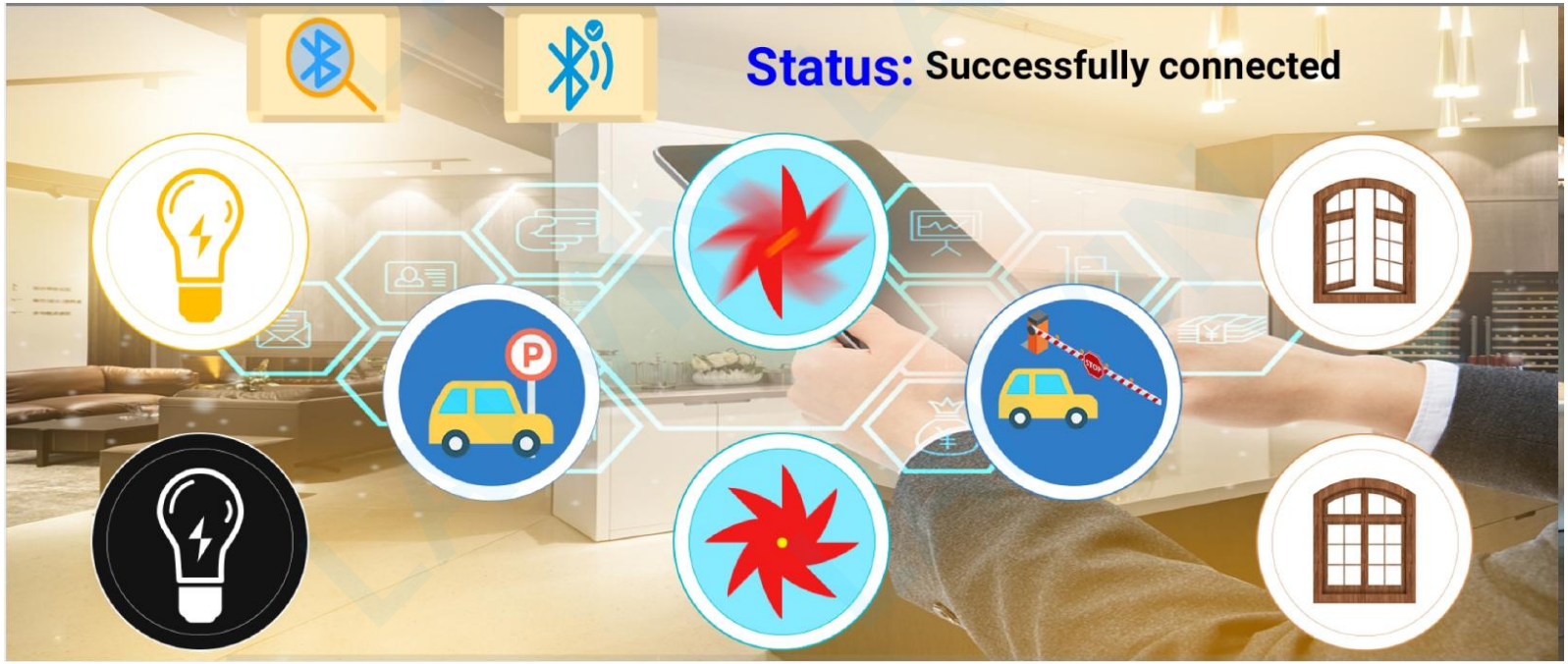


Let's program

Project purpose

Write code to control the realization function:

After the APP is connected to the Bluetooth module, send instructions through the app button to control the turning on and off of the LED and the turning on and off of the fan

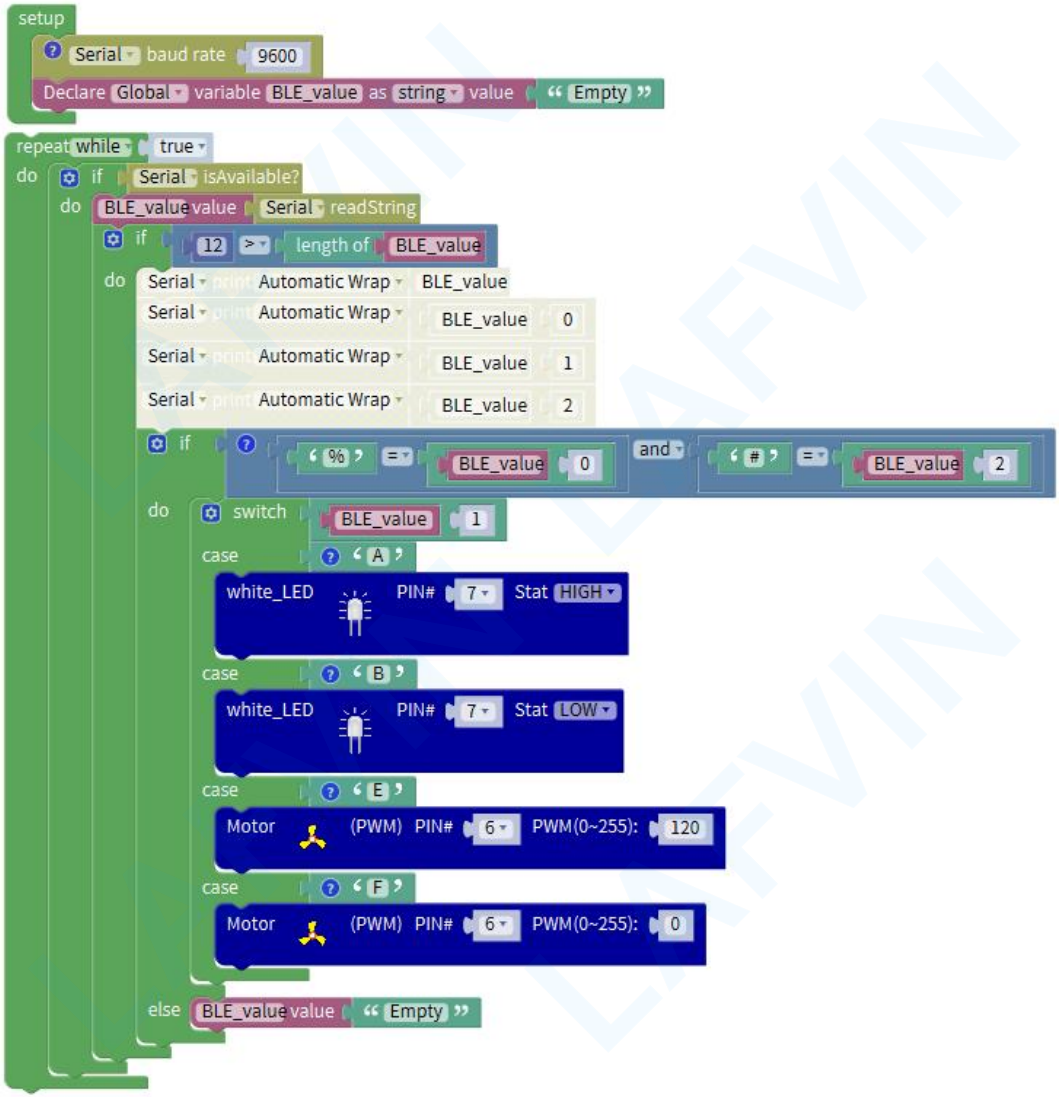


Mixly Code

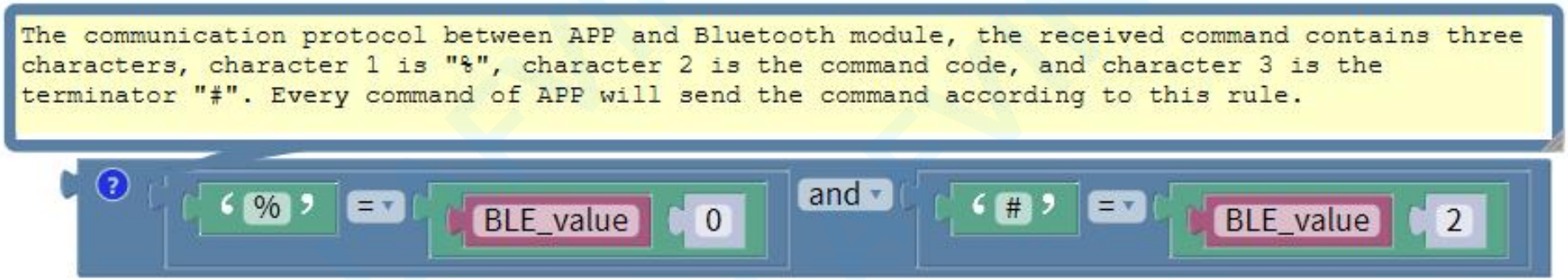
Wire it up well as the above diagram. Okay, let's move on to write the test code.Open Mixly software.Click "**New**" to add new project, then start your programming .

if you want to refer to the program we provide.open the reference code for this project"**Bluetooth Test.mix**" in the reference materials we provided.

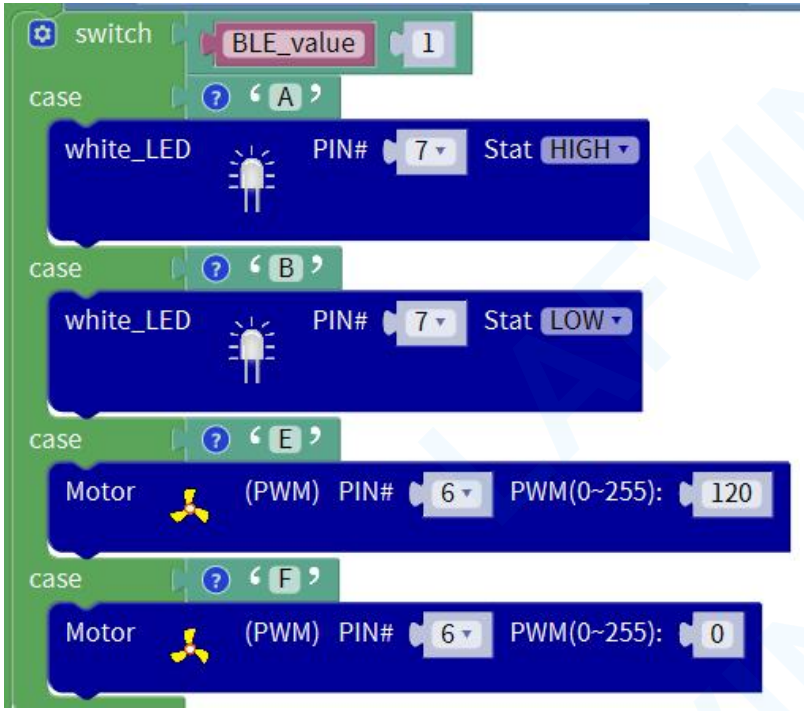
(Note: Before uploading the code, you need to disconnect the Bluetooth module from the sensor shield v5. After successfully uploading the code, reinstall the Bluetooth module.)


setup
Serial baud rate 9600
Declare Global variable BLE_value as string value "Empty"
repeat while true
do if Serial isAvailable?
do BLE_value value Serial readString
if 12 > length of BLE_value
do Serial Automatic Wrap BLE_value
Serial Automatic Wrap BLE_value 0
Serial Automatic Wrap BLE_value 1
Serial Automatic Wrap BLE_value 2
if '%' = BLE_value 0 and '#' = BLE_value 2
do switch BLE_value 1
case 'A'
white_LED PIN# 7 Stat HIGH
case 'B'
white_LED PIN# 7 Stat LOW
case 'E'
Motor (PWM) PIN# 6 PWM(0~255): 120
case 'F'
Motor (PWM) PIN# 6 PWM(0~255): 0
else BLE_value value "Empty"

Programming Thinking



The communication protocol between APP and Bluetooth module, the received command contains three characters, character 1 is "%", character 2 is the command code, and character 3 is the terminator "#". Every command of APP will send the command according to this rule.



The command to turn on the LED is "%A#"

The command to turn off the LED is "%B#"

The command to turn on the fan is "%E#"

The command to turn off the fan is "%F#"

Arduino Code

If you want to refer to the program we provide.Open the reference code for this project "**Bluetooth Test.ino**" in the reference materials we provided.

(Note: Before uploading the code, you need to disconnect the Bluetooth module from the sensor shield v5. After successfully uploading the code, reinstall the Bluetooth module.)

Programming Thinking

```
/*
LAFVIN Smart Home Kit for Arduino
Project 9
*/
String BLE_value;
```

```
void fan_motor_pwm(int speedpin, int speed)
{
    if (speed <= 0)
    {
        analogWrite(speedpin, 0);
    }
    else if (speed > 255)
    {
        analogWrite(speedpin, 255);
    }
    else
    {
        analogWrite(speedpin,speed);
```

```
    }
}

void setup(){
    Serial.begin(9600);
    BLE_value = "Empty";
    // The baud rate of serial communication is set to 9600
    pinMode(7, OUTPUT);
    pinMode(6, OUTPUT);
    digitalWrite(6, LOW);
}

void loop()
```

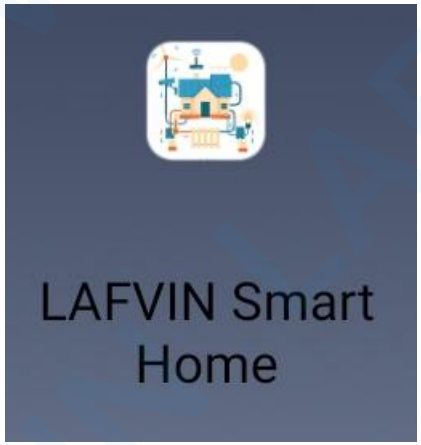
```
{
    if (Serial.available() > 0)
    {
      BLE_value = Serial.readString();
      if (12 > String(BLE_value).length())
      {
        // he communication protocol between APP and Bluetooth module, the received command contains three characters, character 1 is "%", character 2 is the command code, and character 3 is the terminator "#". Every command of APP will send the command according to this rule.
        if ('%' == String(BLE_value).charAt(0) && '#' == String(BLE_value).charAt(2))
        {
          // The command to turn on the LED is "%A#"
          // The command to turn off the LED light is "%B#"
```

```
// The command to turn on the fan is "%E#"
// The command to turn off the fan is "%F#"
switch (String(BLE_value).charAt(1)) {
  case 'A':
    digitalWrite(7,HIGH);
    break;
  case 'B':
    digitalWrite(7,LOW);
    break;
  case 'E':
    fan_motor_pwm(6, 120);
    break;
  case 'F':
```

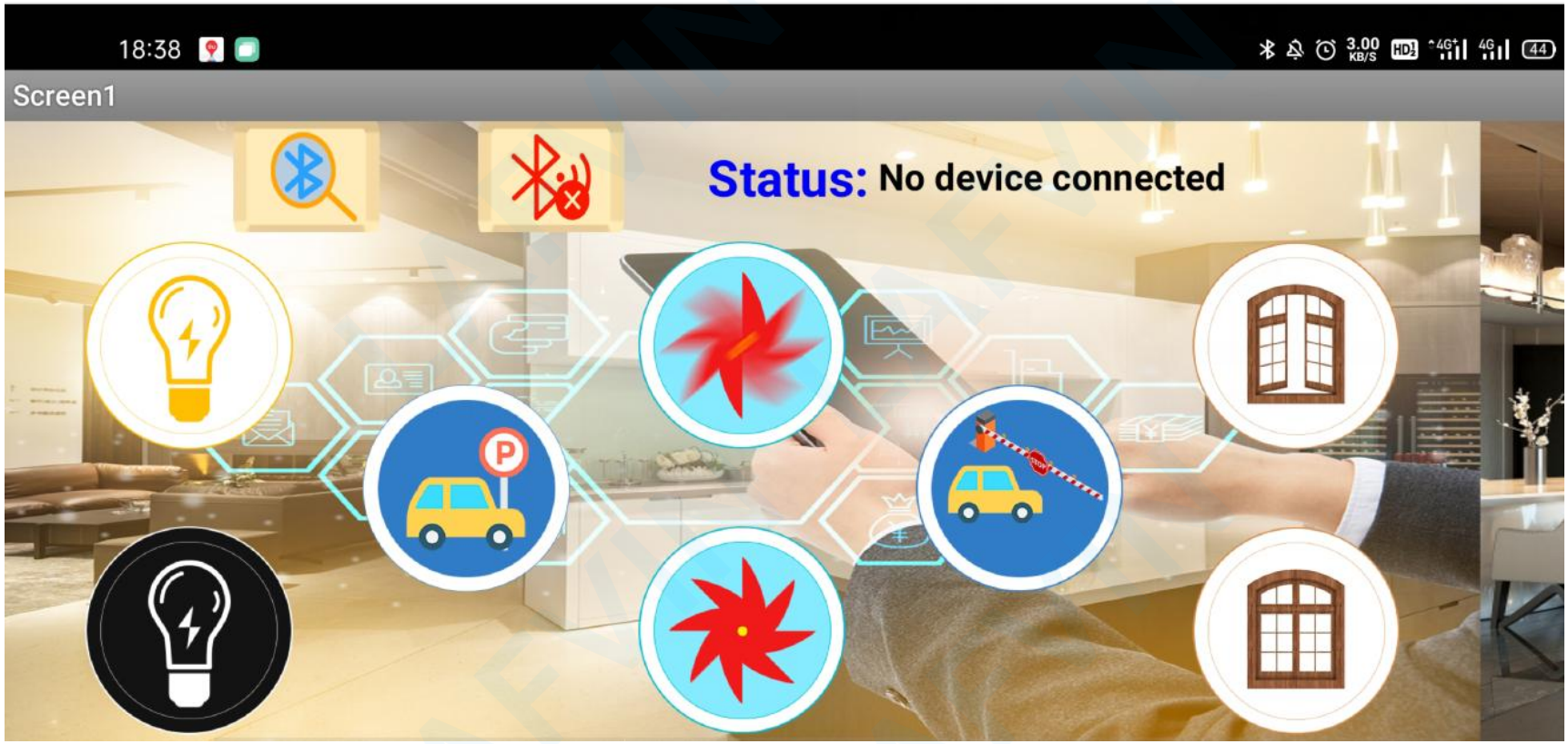
```
                fan_motor_pwm(6, 0);
                break;
            }
        }
        else
        {
            BLE_value = "Empty";
        }
    }
}
}
```

How to connect APP with Bluetooth module

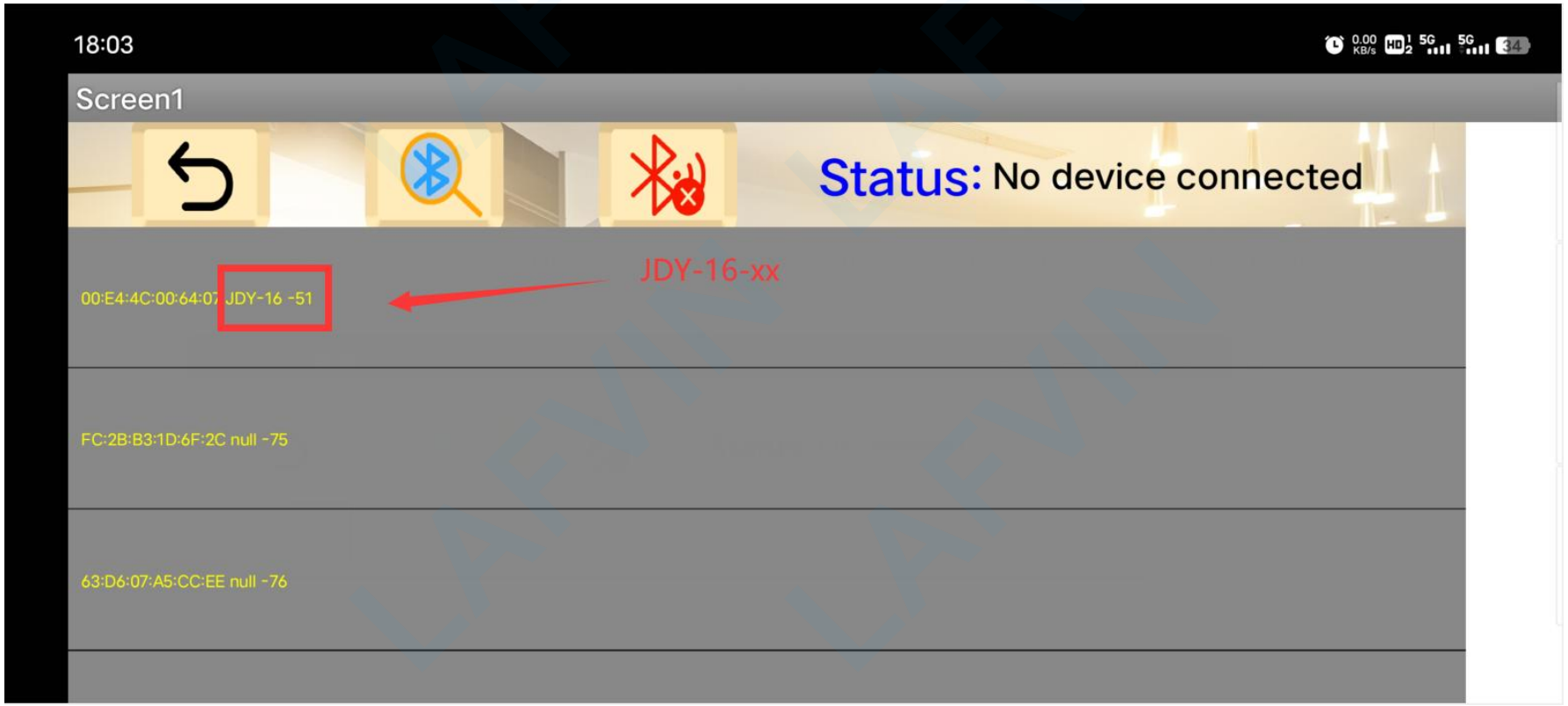
Firstly, download the “Smart_Home_Kit_LAFVIN.apk” file from the folder to your mobile phone and install it into an application software.



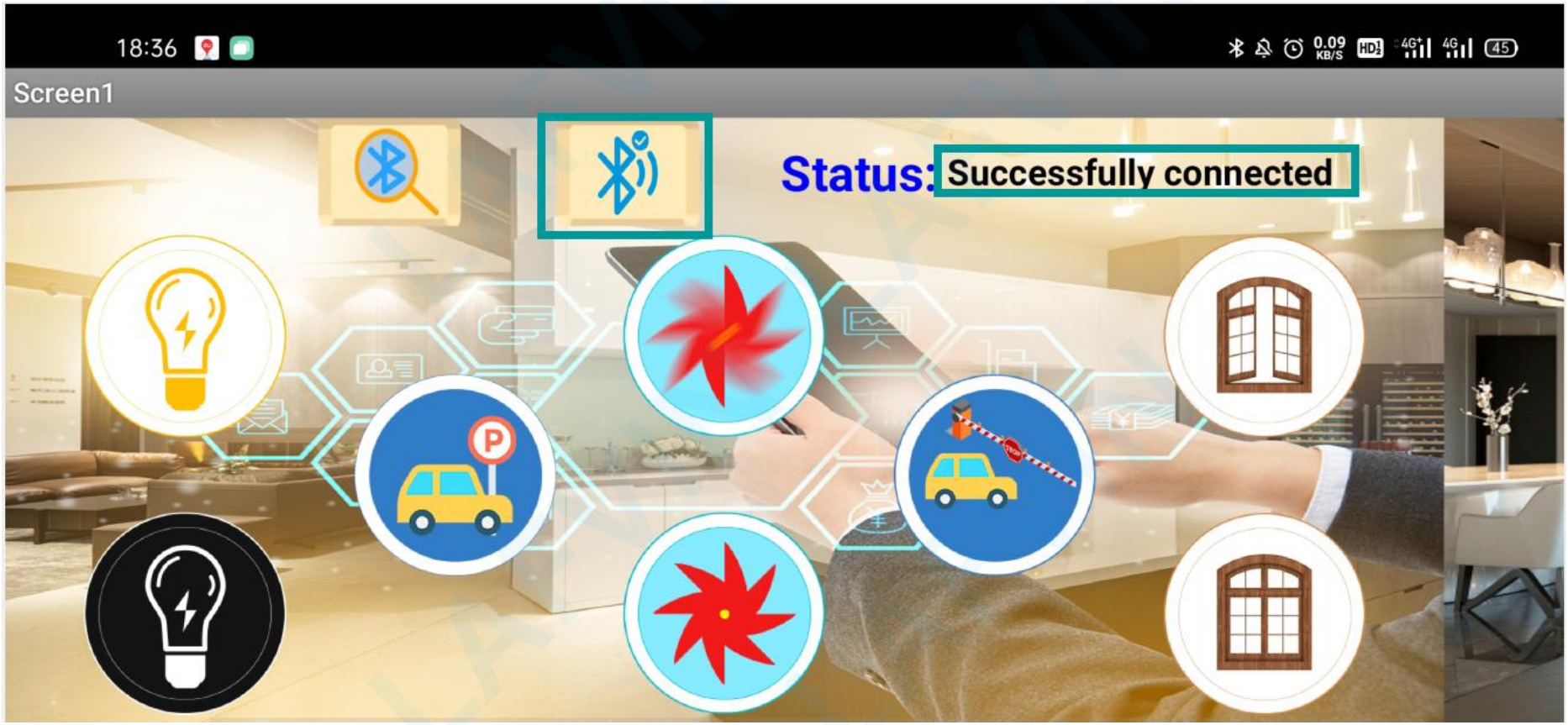
Open the app, you will see the following interface, swipe left and right to switch pages.



After uploading the code to the arduino uno correctly, install the bluetooth module, and then click device scan  ,After successfully searching for the device **JDY-16**, click the device address in the device list to automatically connect.



After successful connection, an icon and status prompt will appear. If the connection is not successful, check the error according to the status prompt, and search for the device again to connect



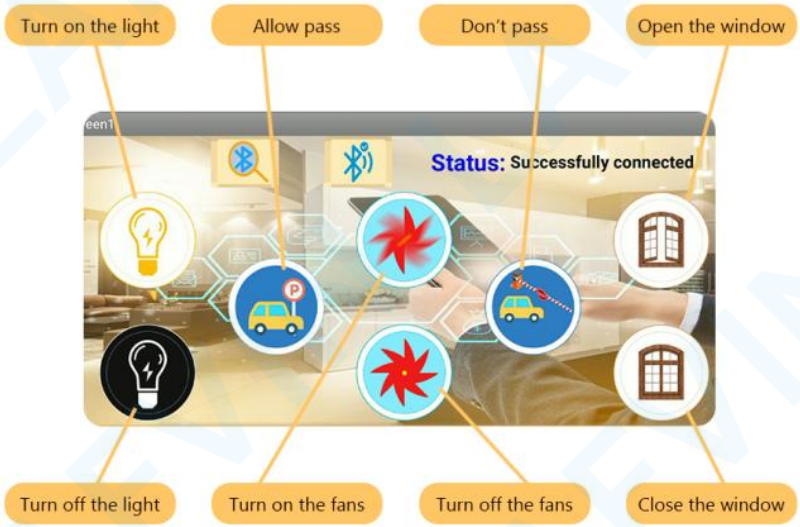
Swipe right to enter the sensor data monitoring interface,

Swipe right to enter the APP password opening function interface

Test Result:

Before uploading the code, you need to remove the Bluetooth device from the Arduino Sensor shield V5.0. After successfully uploading the code, reinstall the Bluetooth device, then open the APP, search for the JDY-16 device to connect, and an icon will appear after successful connection 

. Then you can test the following functions:



(Note: Before uploading the code, you need to disconnect the Bluetooth module from the sensor shield v5. After successfully uploading the code, reinstall the Bluetooth module.)

Project 10：Multi-purpose Smart Home on APP

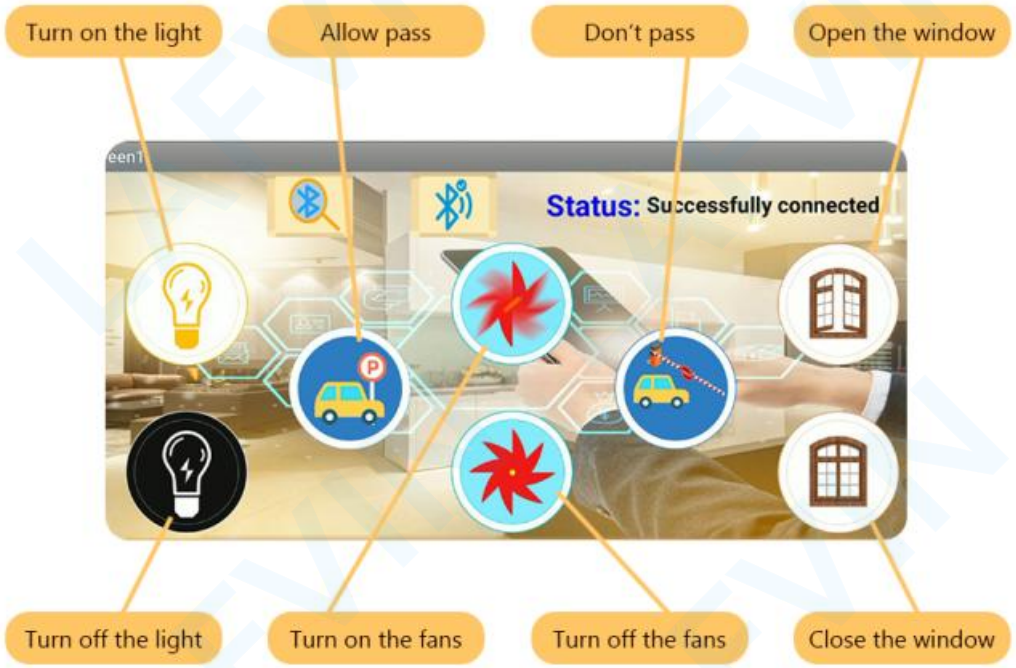
Overview

In the previous project, we learned that all functions are executed at the same time, and also tested the wireless remote control function of the app. In order to make the smart home smarter, we need to add the function of wireless remote control to the multi-function simultaneous execution program. In this way, we can control all the devices through APP, including doors, windows, parking gates, LED modules, fan modules, and monitor the data of all sensors.

The wireless control of multifunctional smart home on APP has the following characteristics:

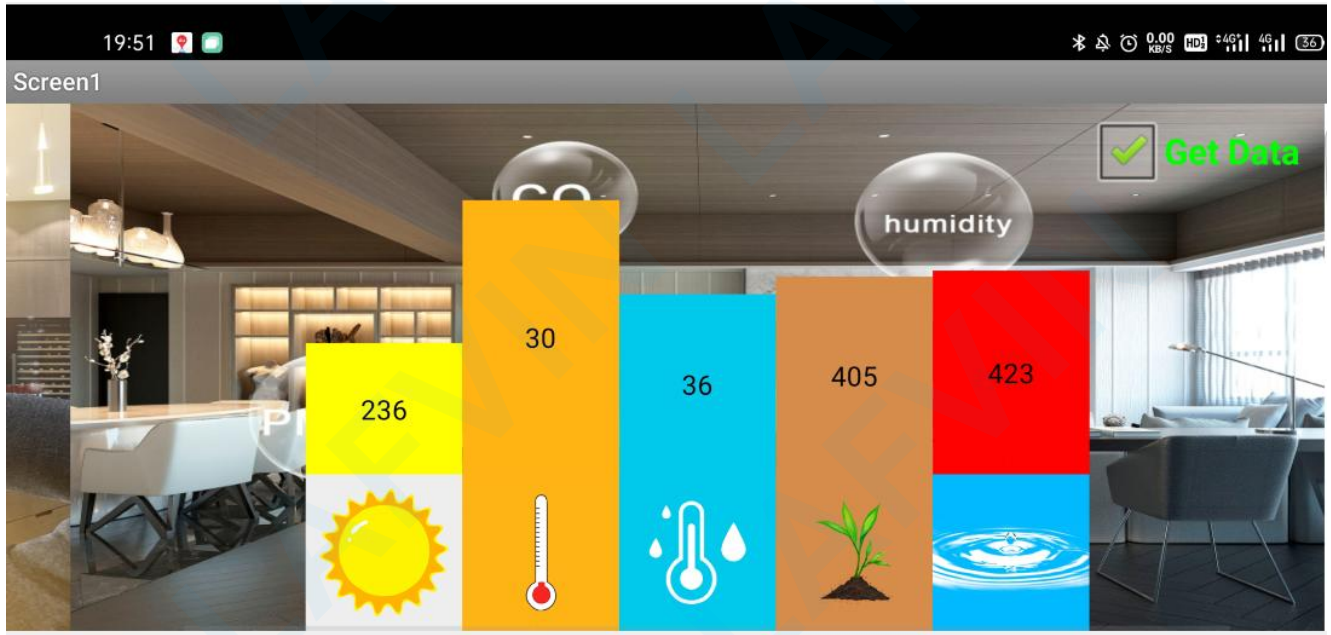
Actuator control interface:

The following buttons can be used to turn on and off the equipment at home at any time



Sensor data monitoring interface:

Turn on the "Get Data" switch, and the monitoring interface will get the latest sensor data every 500 milliseconds. The data obtained are related to the intensity of illumination (0~1024), ambient temperature (0-50 degrees Celsius), ambient humidity (20-90%), soil humidity (0-1024), and water level (0-1024).



Password to open the door interface:

The door open password set in the program is: 1234. You can enter 1234 in the password input box and click the open button. The smart home terminal can return the door open status to the APP. If the door is opened successfully, the APP interface will display "Right". If opening the door fails, the APP interface will display "Error". Similarly, after opening the door successfully, you can click the close button on the APP interface to close the door.

Similarly, in the APP wireless control module, the offline functions of the home can also be executed at the same time. For example, when you monitor the sensor data change in the app, it just happens to be rainy and the appx shows that the water level data increases, and the rain is also good. The controller executes the command to close the window.

The following functions can be executed simultaneously with APP remote control.

(1) Sound Photosensitive Control LED.

Contains light sensor, sound sensor module and LED. When

At night, when someone passes by and makes the sound of footsteps, the LED lights up; no one is nearby,

The indicator light goes out.

(2) Coin Parking

Contains Photoelectric interrupt Module, Servo.

When a coin is inserted, the parking gate will open.

when the coin is taken out, the parking gate will be closed.

(3) Rain-Controlled Window

Contains Water Lever Detection Sensor Module, Servo.

the initial state of the window is open. The water level detection sensor is installed on the top of the house. When it rains, the sensor can directly detect the change of rainwater.

When it is raining, the water level detection sensor detects that the corresponding analog voltage value of the rainfall is greater than 100 and the window will be closed.Otherwise the window is open

(4) Plant Watering Warning

Including Soil Humidity Sensor, Passive Buzzer, LED Module.

When the soil moisture sensor detects that the soil moisture value is less than 50, the sound and light alarm is activated, the red LED light of the sound and light alarm will flash, and the buzzer will emit a ticking sound.

Note: The soil moisture sensor must be inserted into the soil of the plant. If it is not inserted into the soil, the humidity detected by the sensor is 0, which means extreme drought and the alarm will always sound.

(5) Flame Alarm

Including Flame sensor, DC motor fan module.

After uploading the code, use a lighter to ignite near the flame sensor, and the fan will start until the flame is blown out before stopping. If the presence of flame is not detected, the fan will not start.

(6) Intelligent Temperature Control Fan

Including Temperature and humidity sensor module, DC motor fan module.

After uploading the code, the temperature and humidity sensor is always in the detection state. When the ambient

temperature is greater than 31 degrees Celsius, the fan starts and starts at a proportional speed of 120 to dissipate the air in the home. The fan will not stop until the ambient temperature is below 31 degrees Celsius.

(7) Password Door

Including Button Module, 1602 LCD Display, Servo.

When the red button is pressed for more than 500 milliseconds, the entered password value is "-"

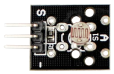
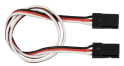
When the red button is pressed for less than 500 milliseconds, the entered password value is "."

In the experimental reference program, we set the password as ".--.".

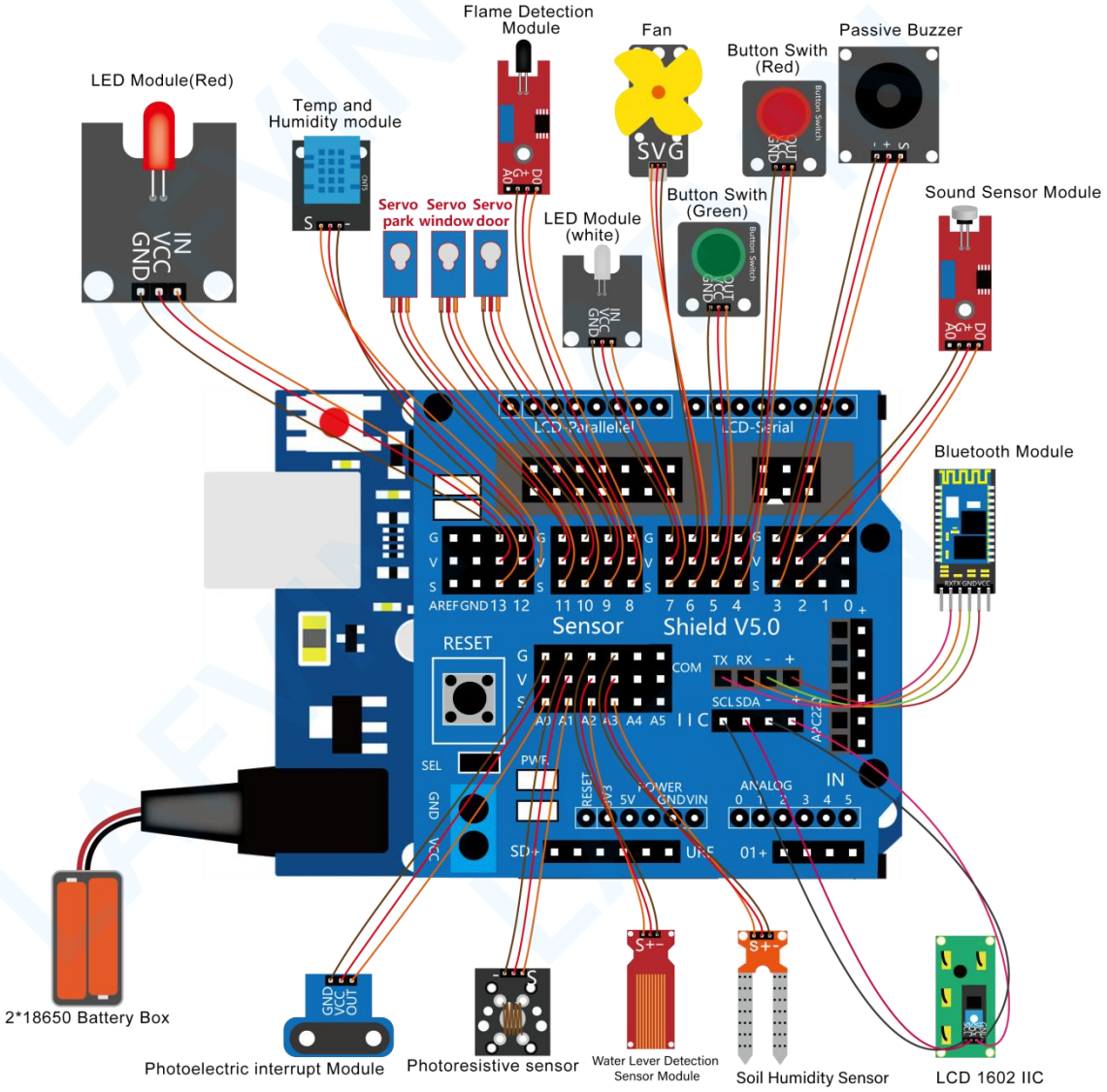
When the password is entered, press the green button to confirm the password. If the entered password is equal to ".--.", the password is correct, and the LCD displays Password: Right. If the password is not equal to ".--.", the password is wrong, and the LCD displays Password: Error. Then wait for one second and the LCD displays Password : Again. At this time, press the red button again to enter the password.

When the password is seriously correct, the door will open. If the green button is pressed again, the door can be closed manually.

Experiment equipment:

Arduino UNO*1	Arduino Sensor Shield V5.0*1	USB cable*1	Button Module*2	Servo*3	1602 LCD Display*1	Flame sensor*1	Fan Module*1
							
Photocell Sensor*1	Sound sensor module*1	LED module*2	Photoelectric Speed Sensor*1	Water Lever Detection Sensor*1	Soil Humidity Sensor*1	Passive Buzzer*1	Temperature and humidity sensor *1
							
2pin F-F Dupont line*2	3pin F-F Dupont line*12	3pin PH2.0 to Dupont line*1	Battery box*1	Bluetooth Module*1			
							

Wiring Diagram：



Let's program

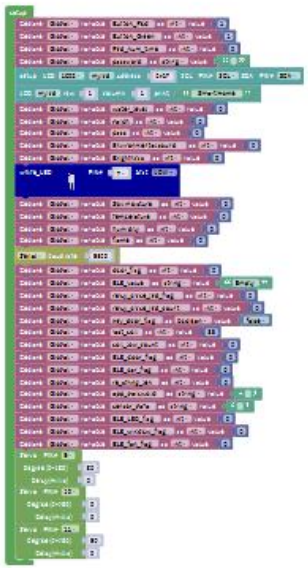
Project purpose

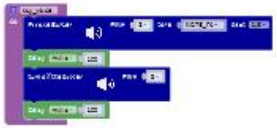
In the experimental reference program, we set the password as ".--.". When the password is entered, press the green button to confirm the password. If the entered password is equal to ".--.", the password is correct, and the LCD displays Password: Right. If the password is not equal to ".--.", the password is wrong, and the LCD displays Password: Error. Then wait for one second and the LCD displays Password: Again. At this time, press the red button again to enter the password.

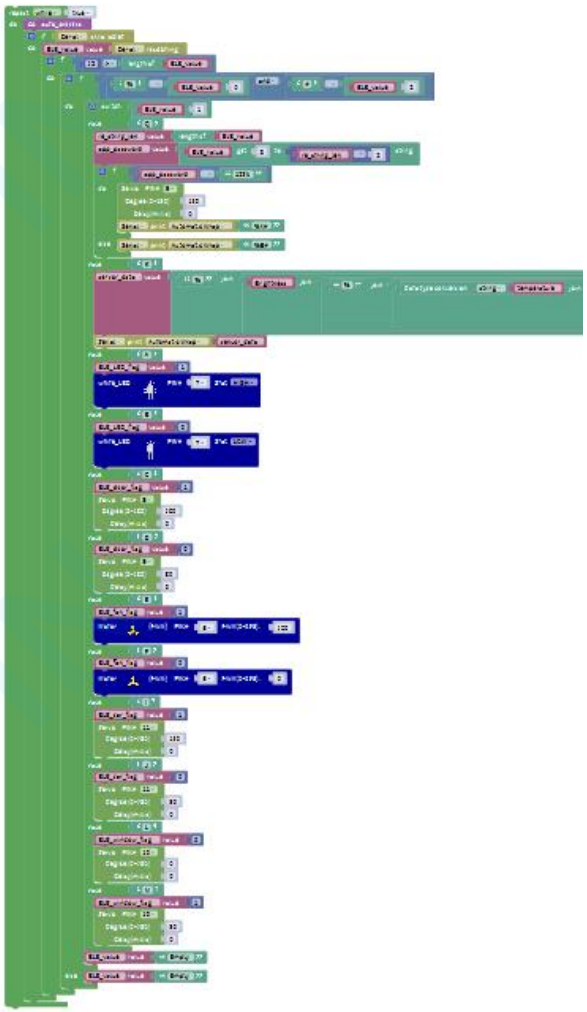
Mixly Code

Wire it up well as the above diagram. Okay, let’s move on to write the test code.Open Mixly software.Click “**New**” to add new project, then start your programming .

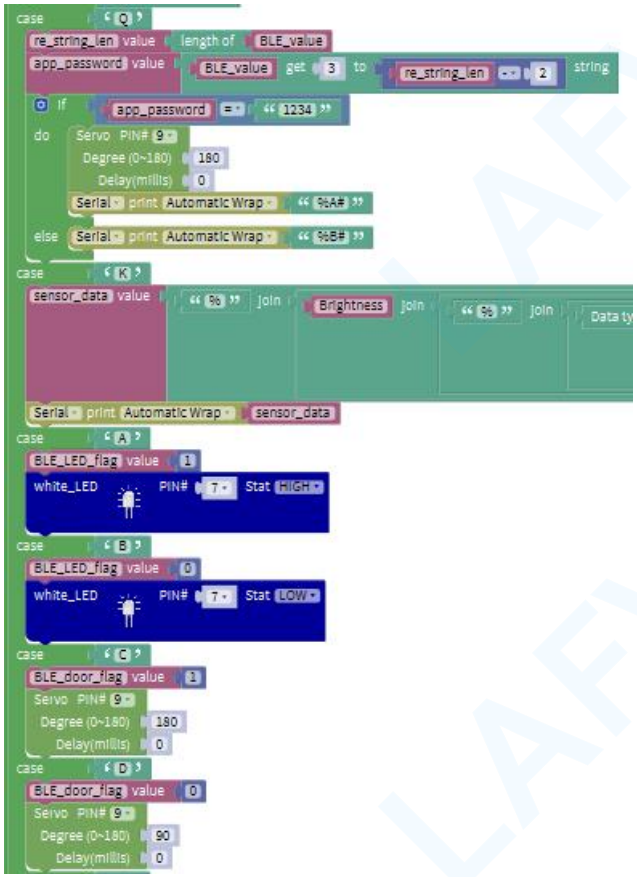
if you want to refer to the program we provide.open the reference code for this project"**Multi-purpose Smart Home On APP.mix**" in the reference materials we provided.







Programming Thinking



The command to open the door is "%Q1234#"

The command to get data is "%K#"

The command to turn on the LED is "%A#"

The command to turn off the LED is "%B#"

The command to close the door is "%D#"

The command to turn on the fan is "%E#"

The command to turn off the fan is "%F#"

The instruction to open the parking gate is "%I#"

The instruction to close the parking gate is "%J#"

The command to open the window is "%M#"

The command to close the window is "%L#"

Arduino Code

If you want to refer to the program we provide.Open the reference code for this project "**Multi-purpose Smart Home On APP.ino**" in the reference materials we provided.

Before you can run this, make sure that you have installed the < Servo> <LiquidCrystal_I2C>library or re-install it, if necessary. Otherwise, your code won't work.For details about loading the library file, see Lesson**"2.3 How to Add Libraries"** about how to add libraries .

Test Result:

After uploading the code, Through app, we can control the various sensors or modules and make smart home to perform the corresponding function.

(Note: Before uploading the code, you need to disconnect the Bluetooth module from the sensor shield v5.0. After successfully uploading the code, reinstall the Bluetooth module.)