

## Lesson 1 Introduction to the UNIIKER

What is the UNIIKER?

Many may not have heard of UNIIKER, but you may be familiar with micro:bit, Raspberry Pi. UNIIKER, is an open-source hardware designed specifically for Python programming education by DFRobot. It is a magical board that boasts Raspberry Pi-level performance and is as easy to use as micro:bit, capable of running complete Python programs!

Let's begin the first lesson of our journey with the UNIIKER!



### Task Objectives

To display text and emojis on the screen of the UNIIKER.





## Knowledge points

1. Getting to know the UNIHAKER
2. Getting to know the 'unihiker' library
3. Learning how to use the 'unihiker' library to write text and add emojis
4. Learning how to display program effects on the UNIHAKER screen

## Material List

### Hardware List:

	
UNIHAKER x1	Type-C & Micro Dual-Use USB Cable x1

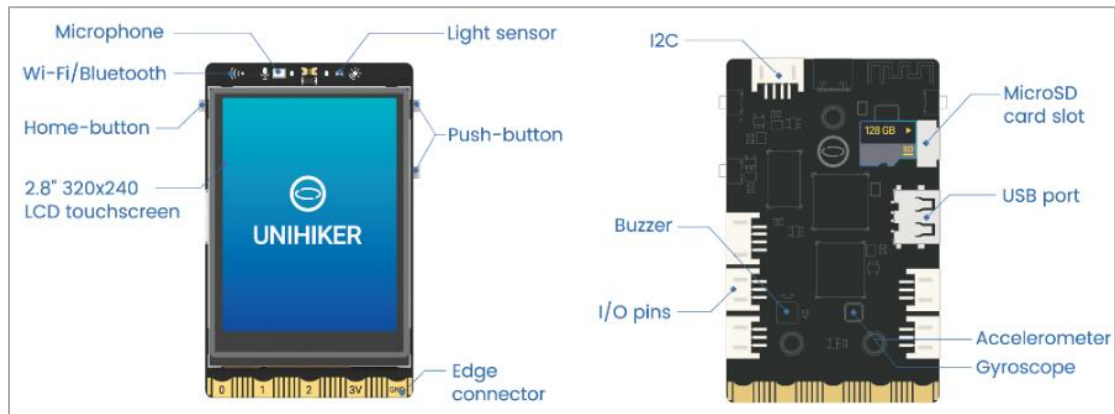
**Software Preparation:** Mind+ Programming Softwarex1

## Knowledge background

1. What is UNIHAKER ?

UNIHAKER, is an open-source hardware designed for educational purposes. It comes with a Linux operating system and Python environment and is pre-installed with commonly used Python libraries, making it capable of handling various programming-related development scenarios, such as building IoT systems, experiencing AI applications, writing electronic games, conducting scientific experiments, designing sound and light interactions, and developing wearable devices.

As an intelligent terminal device, UNIHAKER adopts a microcomputer architecture, integrating an LCD color screen, WiFi Bluetooth, various commonly used sensors, as well as abundant expansion interfaces. all of which are shown in the figure below:



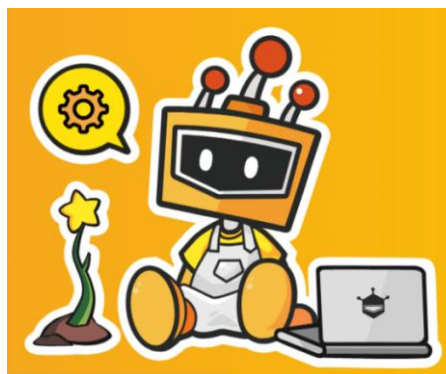
## 2. What is Mind+?

While a physical connection between the UNIHAKER and computer can be established through a USB cable, just doing this is like having a variety of hardware and a computer assembled, but without software to make use of these hardware components. So how can we establish the information connection between these two devices?

The answer is Mind+! It creates a virtual bridge between the two, enabling functions such as code writing and burning, file transfer, and real-time data interaction.

Mind+ is a proprietary programming software designed for teenagers. It integrates various mainstream control boards and over a hundred open-source hardware, supporting artificial intelligence (AI) and the Internet of Things (IoT) functions. It allows users to easily experience the joy of creation, whether by dragging graphical programming blocks or using advanced programming languages like Python/C/C++.

You can download Mind+ at the following address: <https://www.mindplus.cc>.



## 3. What is the unihiker library?

It is a Python library specifically developed for the convenience of using the UNIHAKER board. With the GUI class in the unihiker library, we can achieve screen display and control, such as displaying text, expressions, buttons, etc. Additionally, with the Audio class, we can use the microphone and external speaker to record audio and detect ambient volume, among other functions.

class	Function Description	Detailed Function Explanation
GUI	Screen Control and Display	Display Basic Controls: Display Text, Image, Emoji, Button, Clock, Fill Clock, QR Code  Display Basic Graphics: Display Line, Rectangle, Fill Rectangle, Rounded Rectangle, Fill Rounded Rectangle, Circle, Fill Circle  Detect Mouse and Keyboard: Detect Mouse Coordinates, Keyboard Events  Use Multithreading: Start Threads, Stop Threads
Audio	Microphone and External Speaker Control and Use	Recording: Record for x seconds, start recording, stop recording, get ambient volume.  Audio playback: Play audio file, start playback, pause playback, resume playback, stop playback, get remaining playback time

#### 4. Method for importing the unihiker library's GUI class

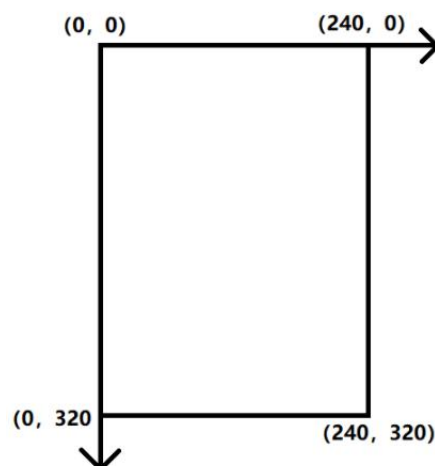
When using the GUI class from the unihiker library to implement functions, we need to first import this module from the library and create an object through instantiation of the class.

```
from unihiker import GUI # Import the GUI module from the UniHiker library
gui = GUI() # Instantiate the GUI class and create an object
```

#### 5. General Knowledge and Functionality of the unihiker Library's GUI Class

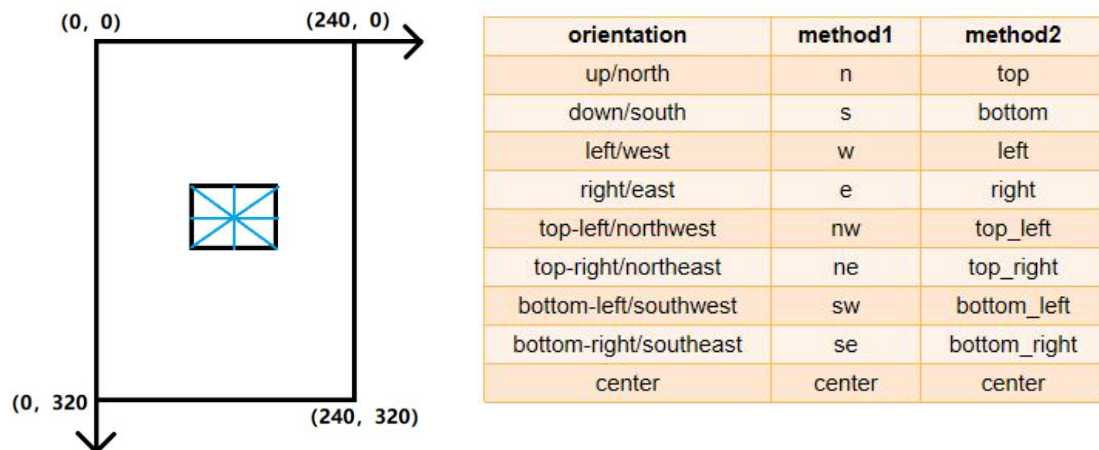
##### (1) Coordinate System

The unihiker board screen has a resolution of 240 x 320, with the coordinate origin located at the top left corner of the screen. The x-axis extends to the right, while the y-axis extends downwards. To display graphical elements at specific locations on the screen, we can program by setting the x and y coordinates accordingly.

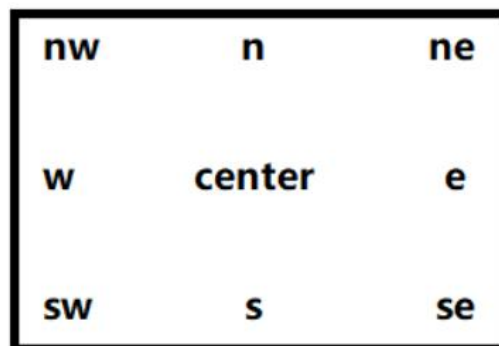


## (2) Alignment Position (Origin)

All the elements on the UNIIKER screen can be referred to as widgets. We can determine their position on the screen by setting their x and y coordinates. In order to facilitate the determination of the widget's relative position within itself, we have set 9 alignment position points for the widget itself, which can be identified using the methods of East, South, West, North and Top, Bottom, Left, Right.



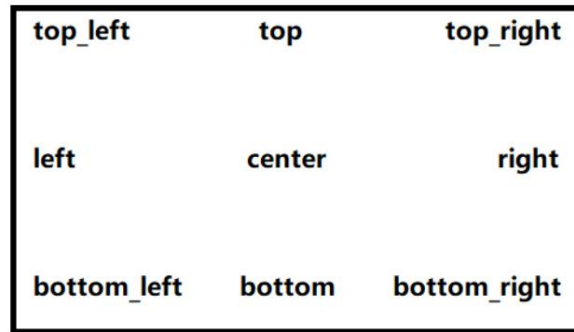
**Method 1:** is to use cardinal directions (north, south, east, and west) as alignment points.



For example, to align the text "UNIIKER" to the south using the north of the control as the reference point, the position can be set using the command "origin='n'".

```
gui.draw_text(x = 120,y=160,text="UNIIKER",origin='n' )
```

**Method 2:** is to use the relative position on the screen (up, down, left, right) as the reference point.



For example, if we want to align the text "UNIHAKER" to the bottom using the top of the control as the reference point, the position can be set using the command "origin='top'".

```
gui.draw_text(x = 120,y=160,text="UNIHAKER",origin='top' )
```

#### 6. Common methods in the unihiker library's GUI class

There are many methods in the GUI class, and we only use a part of them. To implement the functions in programming, we use the format "object.method()" to call them.

##### (1) The draw\_text() method displays text

The draw\_text() method in the GUI class can display text on the UNIHAKER screen.

```
gui.draw_text(x=30, y=88, color="red", text="You", font_size=20, origin='top_left') # Display the text "You" at coordinates (30, 88) with a red color, font size of 20, and aligned to the top left.
```

In this case, the parameters x and y represent the horizontal and vertical position where the text will be displayed, respectively. The parameter text refers to the content of the text to be displayed, font\_size indicates the size of the font, origin represents the alignment of the text, with the default being the top-left corner, and color indicates the color of the font. Not all of these parameters are required, only add them when necessary. Additionally, there are three different ways to specify the color of the font.

For example, to set the font color to red, you can use any of the following methods:

- Setting RGB values: color = (255, 0, 0)
- Setting hexadecimal values: color = "#FF0000"
- Setting a predefined color: color = "red"

##### (2) The draw\_emoji() method displays emojis

The draw\_emoji() method in the Gui class can be used to display emojis on the UNIHAKER screen.

```
gui.draw_emoji(x=120, y=230, w=100, h=100, emoji="Wink", duration=0.1, origin="center") # displays the built-in emoji "Wink" at the position (120, 230) with a size of 100x100 pixels. The image switches every 0.1 seconds and is centered on the given position.
```

The parameters x and y respectively indicate the horizontal and vertical coordinates of the emoji display position. The parameters w and h indicate the width and height of the emoji, which are

scaled proportionally according to the smaller side. The parameter "emoji" represents the specific name of the emoji to be displayed, which can use built-in emojis (Angry, Nerve, Peace, Shock, Sleep, Smile, Sweat, Think, Wink). The parameter "duration" represents the interval time for switching images, and the parameter "origin" represents the alignment position, which is set to the top left corner by default.

## Hands-on practice

### Task Description 1: Displaying Text and Emojis

Display text and emojis on the UNIIKER screen.

#### 1. Hardware setup

**STEP 1:** Connect the UNIIKER to the computer via a USB cable.

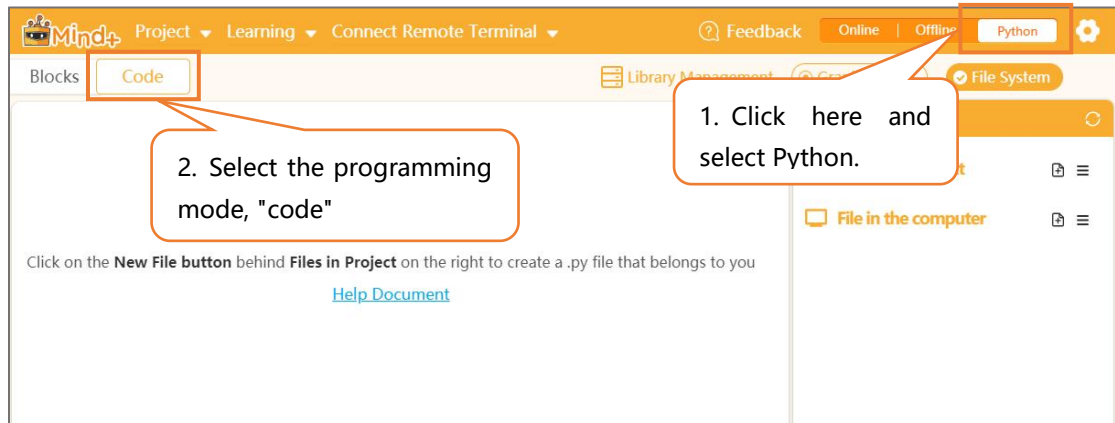


#### 2. program coding

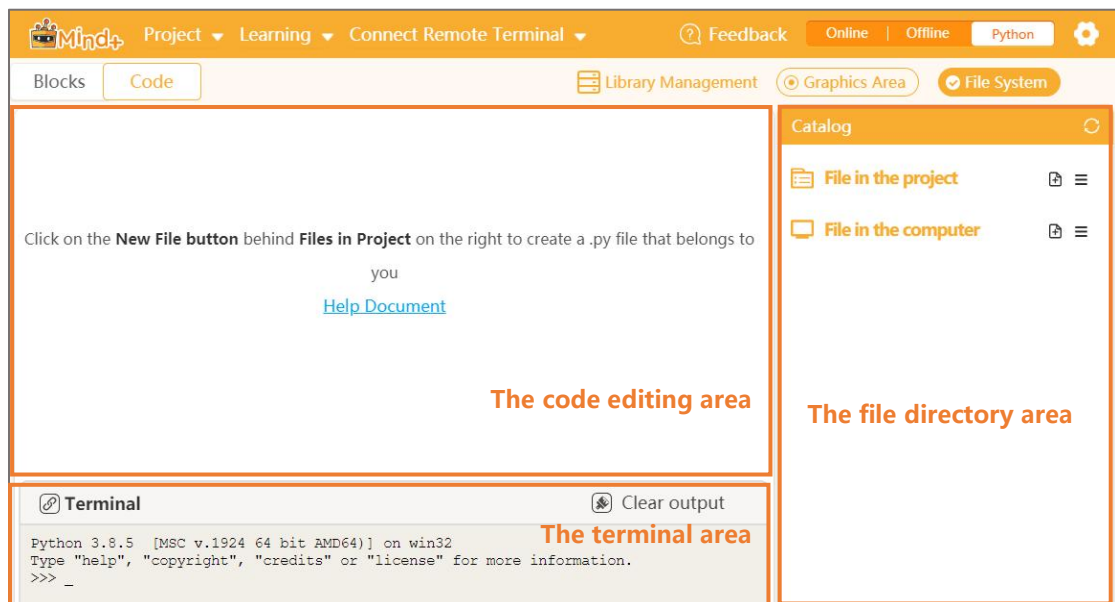
Before writing the program code, we need to create a project file and a Python program file.

**STEP 1:** Creating and Saving Project Files

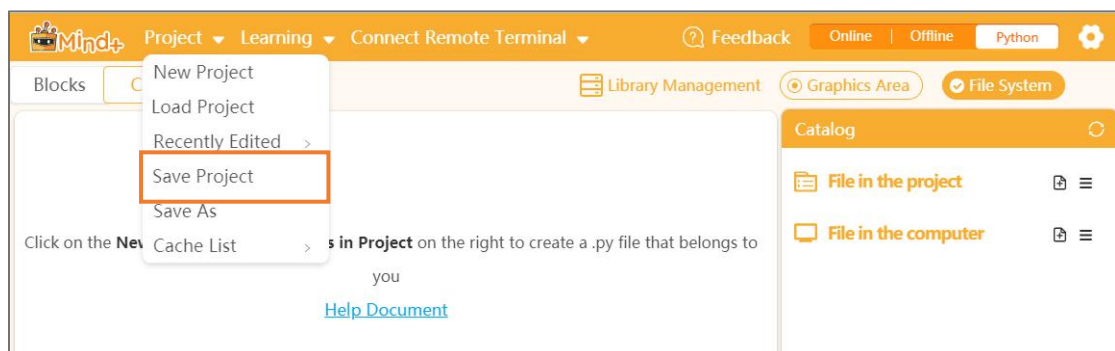
- (1) Launch the Mind+ software, then click the Python button in the upper right corner and switch the programming mode to code.



- (2) After completing the previous step, you will see a screen like this. The right side is the file directory area, and the left side consists of the code editing area and the terminal area.



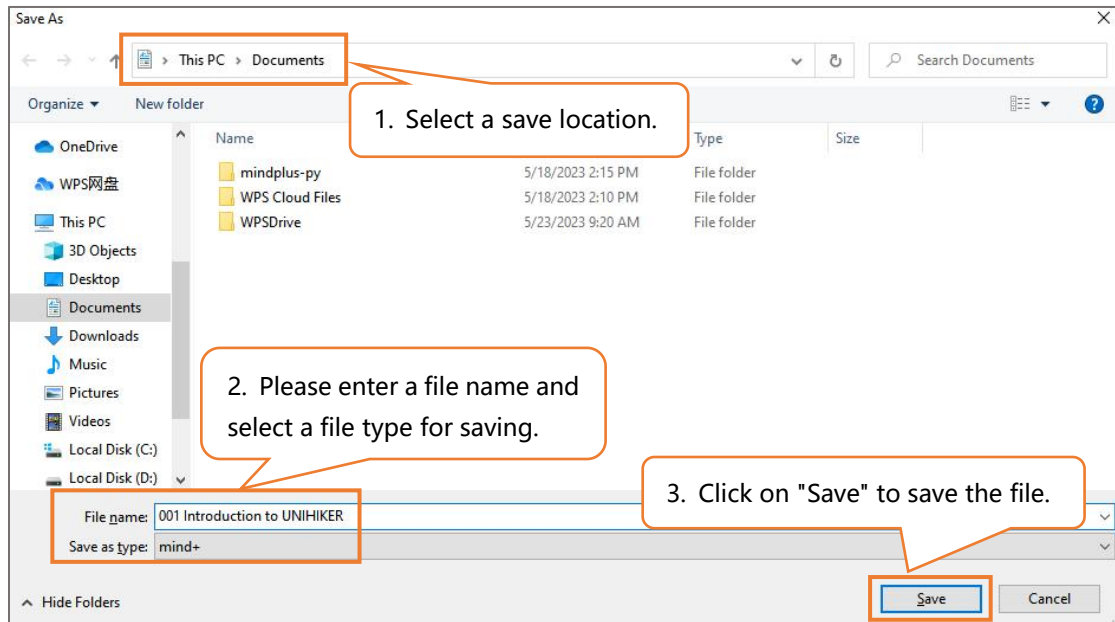
- (3) click on the "Save Project" option in the top-left corner of the "project" menu.



- (4) In the popup window, select the save location, enter the file name "001 Introduction to UNIHAKER", and choose the save type as "Mind+".

**Tips:** You can choose any name for the file.

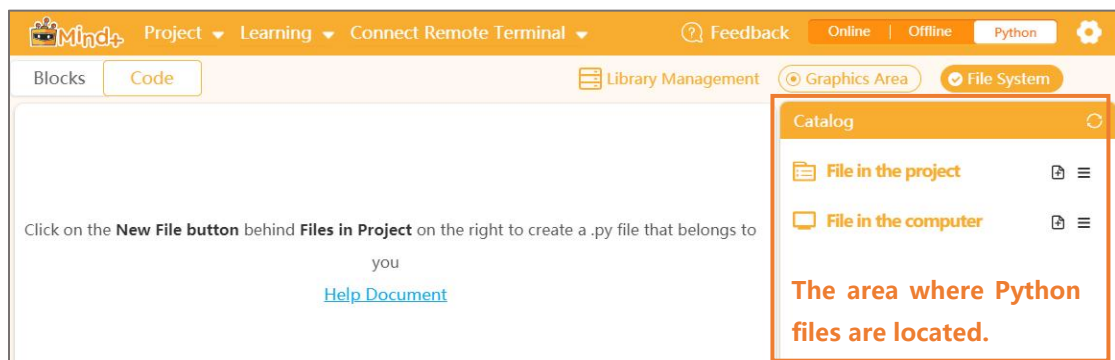




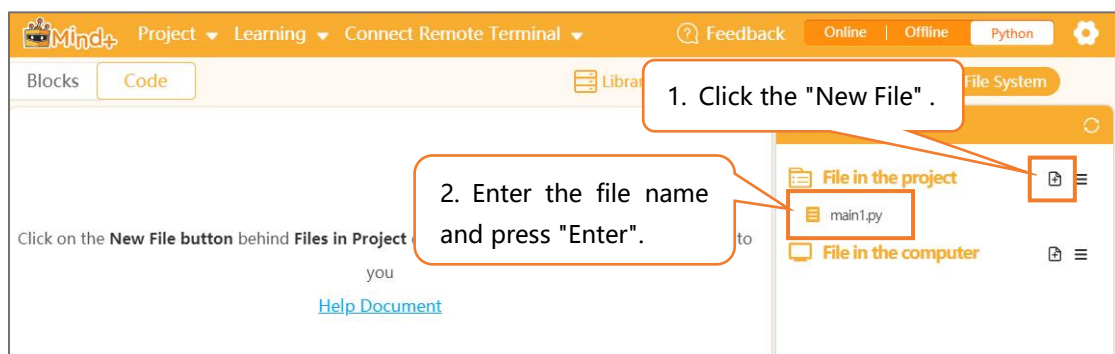
After completing the above four steps, we have successfully created and saved our project file. However, our project is implemented using the Python language. Therefore, next we need to create a Python program file within the project file.

## STEP 2: Creating and Saving Python Files

(1) Locate the file directory area



(2) Create a new Python program file and name it "main1.py"

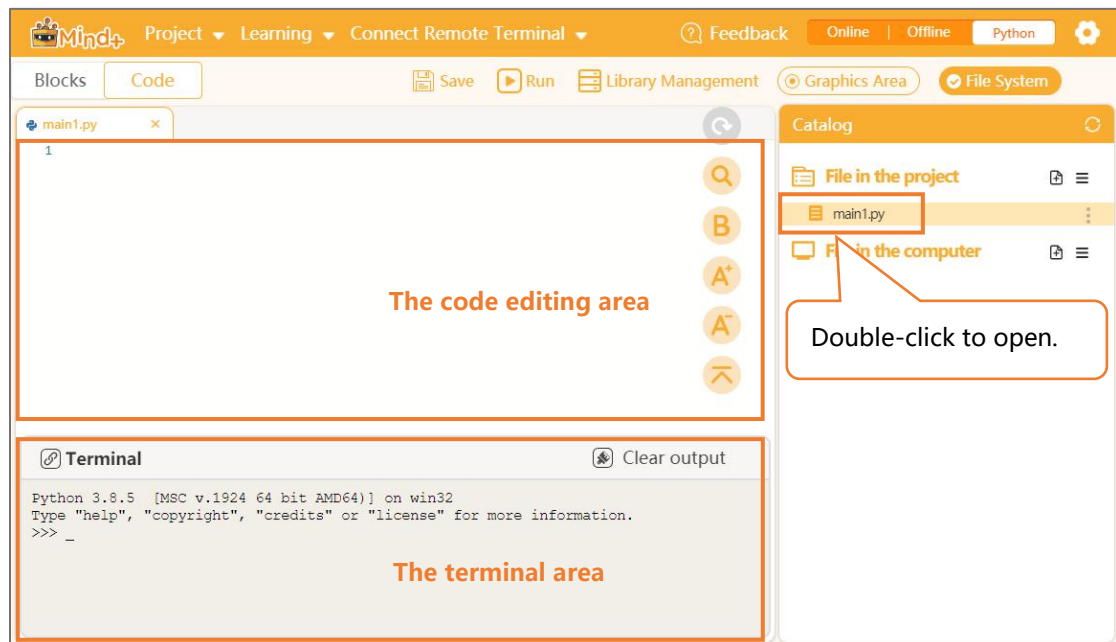


**Tips:** Python program files must end with the ".py" format in order to be programmed, otherwise

they cannot be opened.

### (3) Opening the file

After creating the file, we can double-click on the "main1.py" file to program it in the code writing area of the program, as shown below.



## STEP 4: Programming

### (1) Import the required libraries

For this task, we need to use the GUI module from the unihiker library to display text on the screen. Therefore, we need to import it first. Here is the code for it. Additionally, we also need to import the time library to introduce a certain delay in the program to keep it running.

```
from unihiker import GUI # Import the GUI module from the UniHiker library
import time # Import the time library
```

### (2) Instantiating the GUI class

Before displaying text and emojis using the GUI module from the unihiker library, we need to instantiate the GUI class to create an object that allows us to use the various methods in the class.

```
gui = GUI() # Instantiate the GUI class and create an object
```

### (3) Displaying text

When displaying text on the screen, we can set various properties such as coordinates, color, content, font size, etc. for each text to achieve better effects.

```
gui.draw_text(x=30, y=88, color="red", text="你", font_size=20) # Display the text "你"
# in red color with font size 20 at the coordinates (30, 88)
gui.draw_text(x=60, y=88, color="orange", text="好", font_size=20,) # Display the text "好"
# in orange color with font size 20 at the coordinates (60, 88)
```

```

gui.draw_text(x=90, y=88, color="yellow", text=",", font_size=20) # Display a comma in yellow
color with font size 20 at the coordinates (90, 88)
gui.draw_text(x=120, y=88, color="green", text="行", font_size=20) # Display the text "行"
" in green color with font size 20 at the coordinates (120, 88)
gui.draw_text(x=150, y=88, color="cyan", text="空", font_size=20) # Display the text "空"
" in cyan color with font size 20 at the coordinates (150, 88)
gui.draw_text(x=180, y=88, color="blue", text="板", font_size=20) # Display the text "板"
" in blue color with font size 20 at the coordinates (180, 88)
gui.draw_text(x=210, y=88, color="purple", text="!", font_size=20) # Display an exclamation
mark in purple color with font size 20 at the coordinates (210, 88)

gui.draw_text(x=15, y=150, color=(255,105,180), text="Hello,", font_size=20) # Display the tex
t "Hello," in a custom color with font size 20 at the coordinates (15, 150)
gui.draw_text(x=95, y=150, color=(0,191,255), text="UNIHiker!", font_size=20) # Display the t
ext "UNIHiker!" in a custom color with font size 20 at the coordinates (95, 150)

```

#### (4) Displaying emojis

There are many emojis that can be displayed on the UNIHiker screen. Here, we choose the "Wink" emoji to present it below the text.

```

gui.draw_emoji(x=120, y=230, w=100, h=100, emoji="Wink", duration=0.1,origin="center") #
Display the built-in emoji "Wink" at the coordinates (120, 230), with a size of 100x100 pixels, a
duration of 0.1 seconds for each image switch, and center alignment.

```

#### (5) Keep the content displayed

Finally, in order to keep the screen content displayed for a long time, we run the program continuously.

```

while True: # Loop indefinitely
    time.sleep(1) # Delay for 1 second

```

**Tips:** The complete example program is as follows:

```

from unihiker import GUI # Import the GUI module from the UniHiker library
import time # Import the time library

gui = GUI() # Instantiate the GUI class and create an object

gui.draw_text(x=30, y=88, color="red", text="你", font_size=20) # Display the text "你"
" in red color with font size 20 at the coordinates (30, 88)
gui.draw_text(x=60, y=88, color="orange", text="好", font_size=20,) # Display the text "好"
" in orange color with font size 20 at the coordinates (60, 88)
gui.draw_text(x=90, y=88, color="yellow", text=",", font_size=20) # Display a comma in yellow
color with font size 20 at the coordinates (90, 88)

```

```

gui.draw_text(x=120, y=88, color="green", text="行", font_size=20) # Display the text "行"
" in green color with font size 20 at the coordinates (120, 88)
gui.draw_text(x=150, y=88, color="cyan", text="空", font_size=20) # Display the text "空"
" in cyan color with font size 20 at the coordinates (150, 88)
gui.draw_text(x=180, y=88, color="blue", text="板", font_size=20) # Display the text "板"
" in blue color with font size 20 at the coordinates (180, 88)
gui.draw_text(x=210, y=88, color="purple", text="!", font_size=20) # Display an exclamation m
ark in purple color with font size 20 at the coordinates (210, 88)

gui.draw_text(x=15, y=150, color=(255,105,180), text="Hello,", font_size=20) # Display the text
"Hello," in a custom color with font size 20 at the coordinates (15, 150)
gui.draw_text(x=95, y=150, color=(0,191,255), text="UNIHAKER!", font_size=20) # Display the te
xt "UNIHAKER!" in a custom color with font size 20 at the coordinates (95, 150)

gui.draw_emoji(x=120, y=230, w=100, h=100, emoji="Wink", duration=0.1,origin="center") #
Display the built-in emoji "Wink" at the coordinates (120, 230), with a size of 100x100 pixels, a
duration of 0.1 seconds for each image switch, and center alignment.

while True: # Loop indefinitely
    time.sleep(1) # Delay for 1 second

```

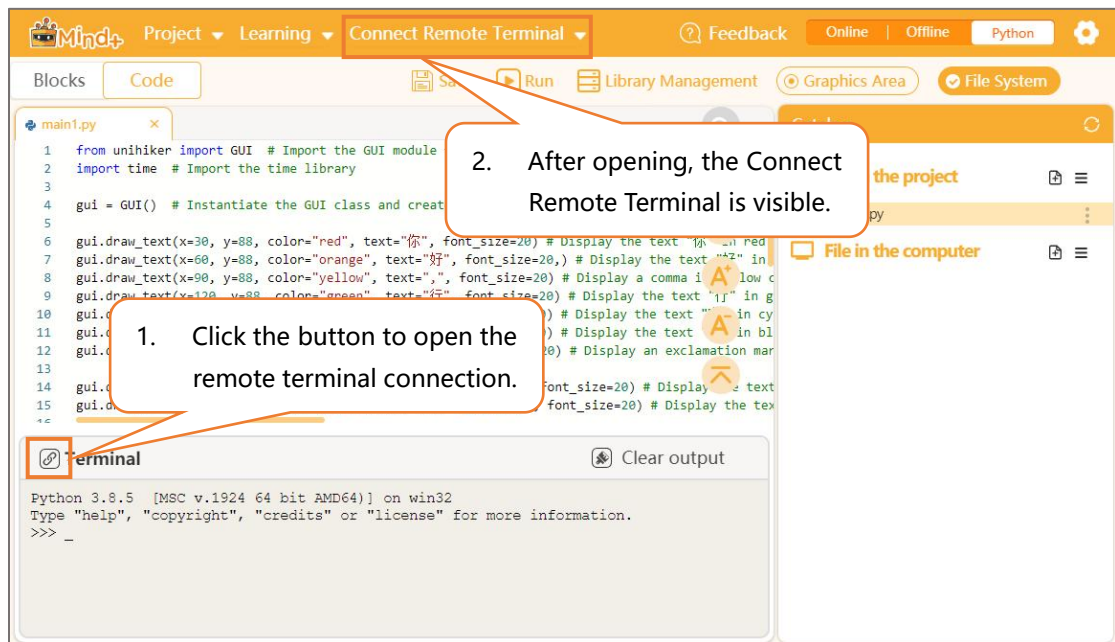
### 3. Running the Program

#### STEP 1: Remote connection to the UNIHAKER

- (1) Connect the UNIHAKER to the computer using a USB cable, and make sure the UNIHAKER is powered on.

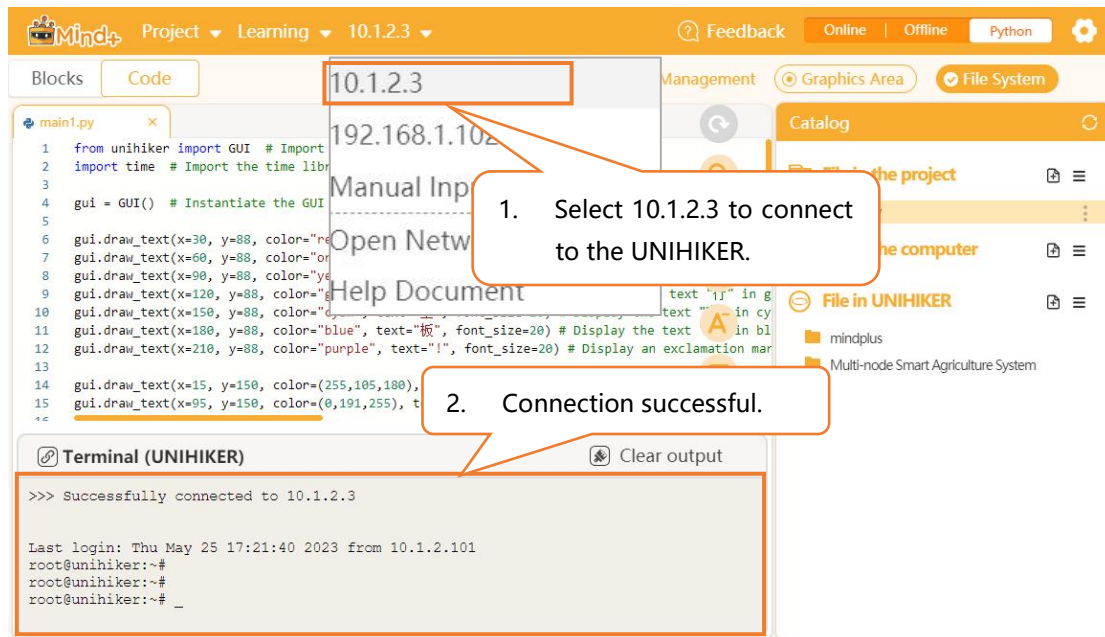


(2) Open the remote connection terminal.

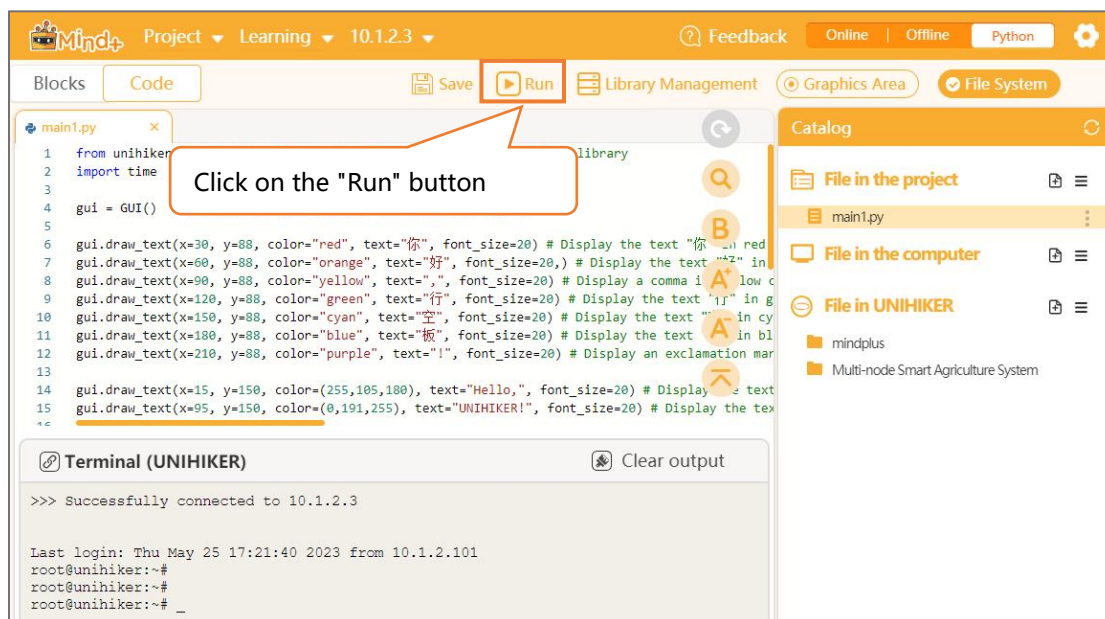


(3) Connect to the UNIHAKER.

**Tips:** "10.1.2.3" is the fixed IP address used for the UNIHAKER board when connected directly to a computer via USB cable.



**STEP 2:** Click on the "Run" button in the upper right corner.



**STEP 3:** Observe the result.

Observe the UNIHAIKER and you will see the words "你好，行空板！" and "Hello, UNIHAIKER!" displayed on the screen, with a "Wink" emoji below the text.



## Challenge Yourself

1. Why not try replacing "Wink" with other emoticons and see how it looks?
2. Do you have any other messages you'd like to display on the line board?