Lesson 10 Button Detection

1. Working Principle

The path to the source code of the program is 5. MaxArm Hardware Basic Learning/Arduino Development/Game Programs/Button Detection/ Key_Detect/Key_Detect.ino.

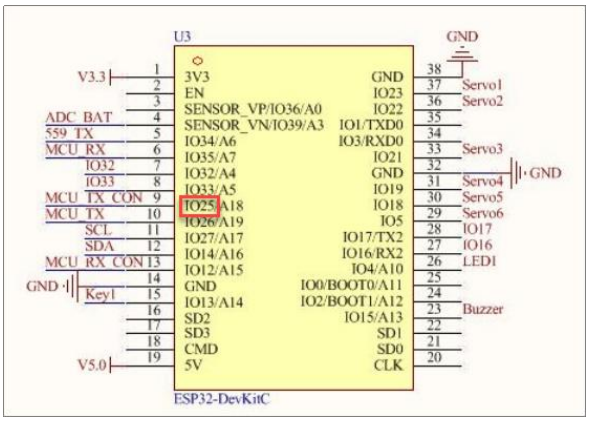
```
int pushButton = 25; // Define button pin

void setup() {
  Serial.begin(9600);
  Serial.println("start...");
  pinMode(pushButton, INPUT); // Set the mode of the button pin as input
}


void loop() {
  int buttonState = digitalRead(pushButton); // Read button value
  if(buttonState == 0){ // When the button is pressed, it is low level
    delay(10); // 10ms delay to remove joggle
    if(buttonState == 0){ Determine again whether the button was pressed
      Serial.println("hello world");
      delay(500);
    }
  }
  delay(10);
}
```

Firstly, define the pin of button, and then read the current button state through digitalRead() function (It is low level when the button is pressed), and determine whether the button is pressed by the judgment statement. Finally, the serial monitor will print “hello world” after the button is pressed.

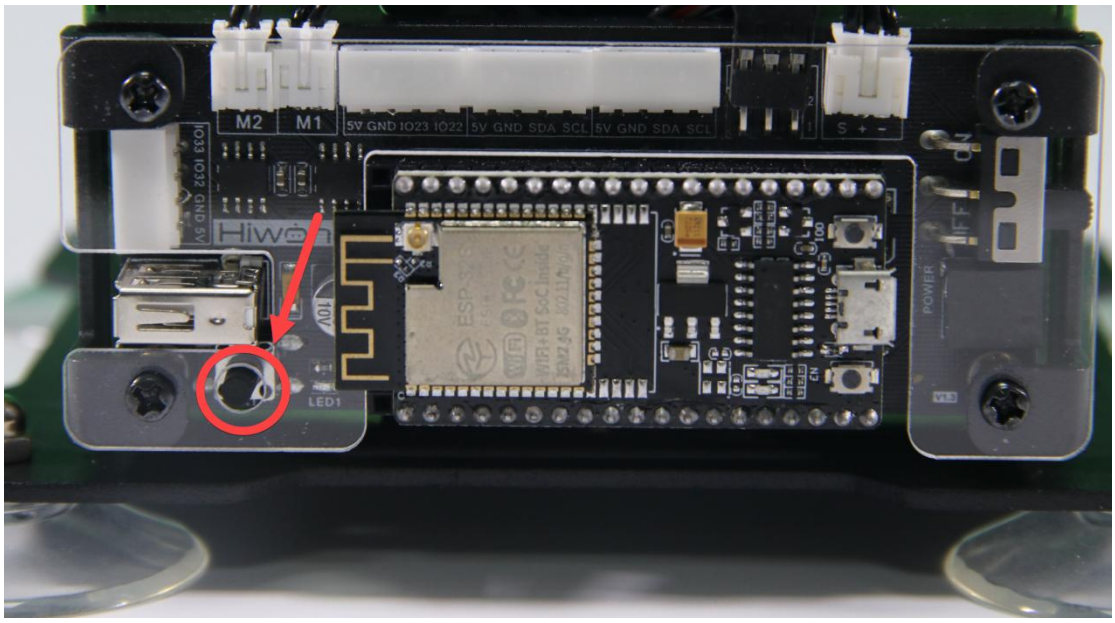
The following figure is the pin information of ESP32 main chip. The K1 key on expansion board is connected to IO25, as shown in the image below.



2. Preparation

2.1 Hardware

There is a K1 key on the main controller of MaxArm, circled in the image below.

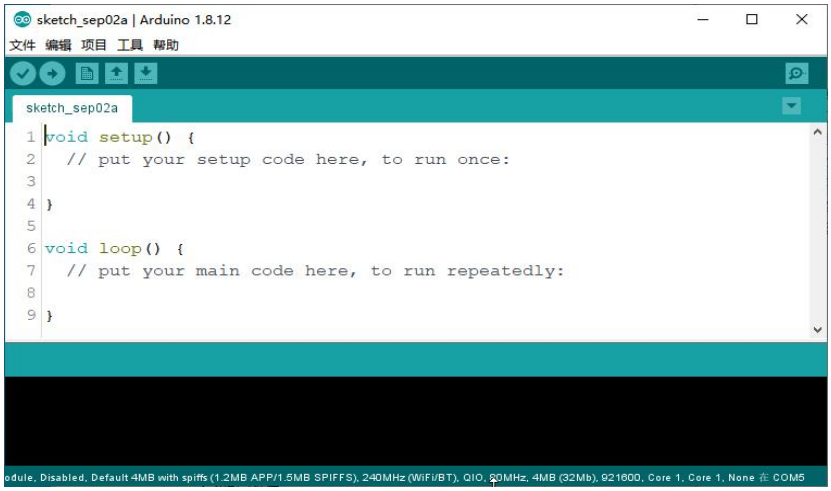


2.2 Software

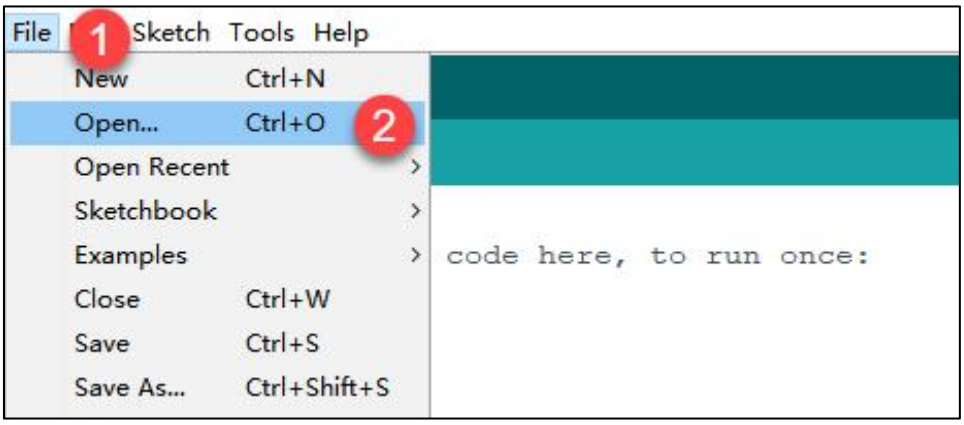
Please refer to the material in folder “4.MaxArm Underlying Program/Lesson 1 Set Development Environment” to connect ESP32 controller to Arduino Editor.

3. Program Download

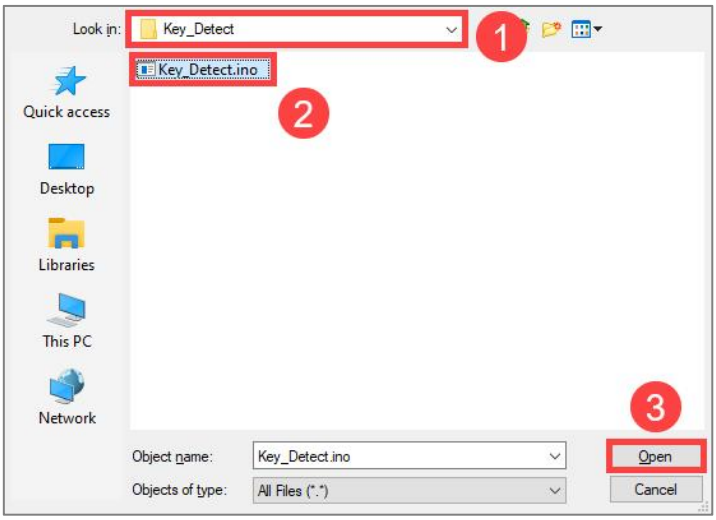
1) Double click to open Arduino IDE.



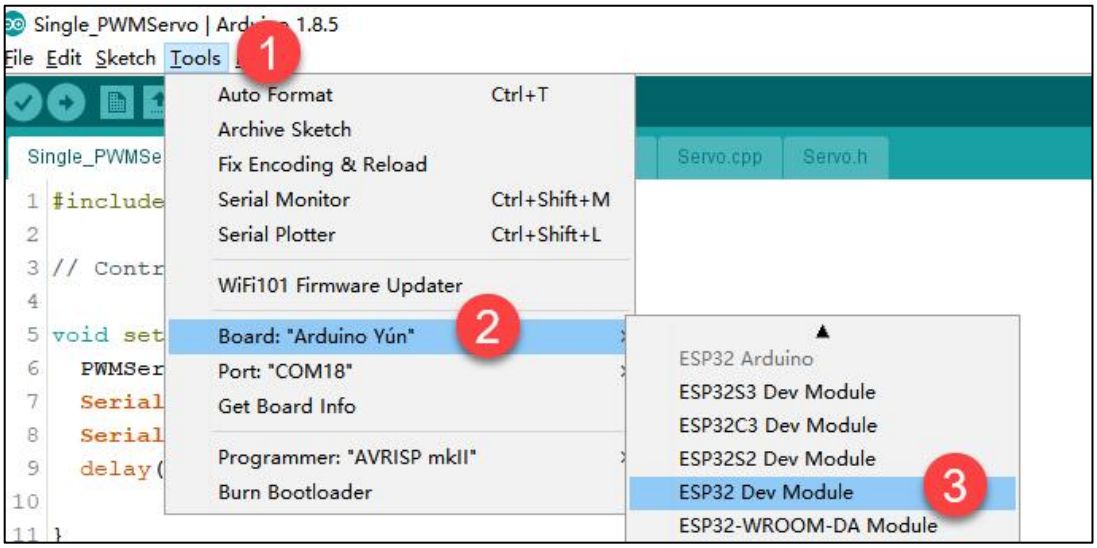
2) Click “File->Open” in turn.



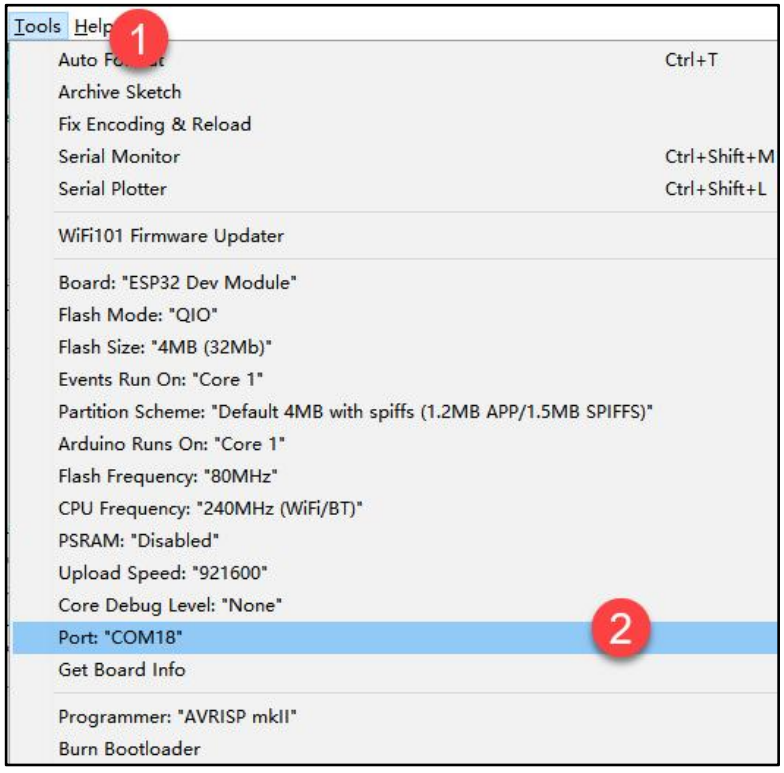
3) Select the program “Key_Detect.ino” in the folder “5.MaxArm Hardware Basic Learning/Arduino Development/Game Programs/Button Detection/ Key_Detect”



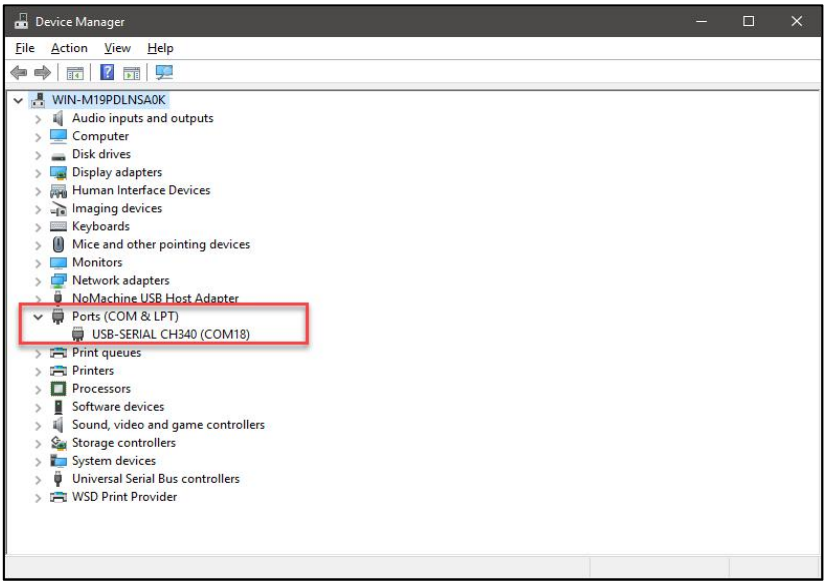
4) Check the board model. Click “Tools->Board” and select “ESP 32 Dev Module”. (If the model of development board has been configured when setting the development environment, you can skip this step.)



5) Select the corresponding port of ESP32 controller in “Tools->Port”. (Here take the port “COM5” as example. Please select the port based on your computer. If COM1 appears, please do not select because it is the system communication port but not the actual port of the development port.)

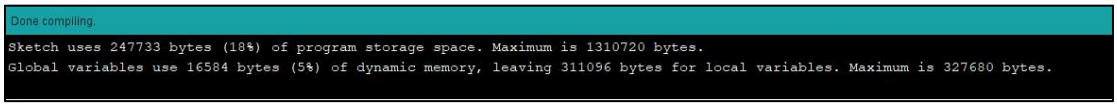


6) If you're not sure about the port number, please open the "This PC" and click "Properties->Device Manger" in turns to check the corresponding port number (the device is with CH340).

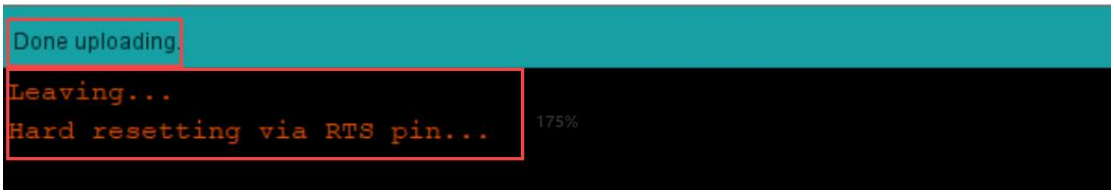


7) After selecting, confirm the board "ESP32 Dev Module" in the lower right corner and the port number "COM5" (it is an example here, please refer to the actual situation).

8) Then click on  icon to verify the program. If no error, the status area will display "Compiling->Compile complete" in turn. After compiling, the information such as the current used bytes, and occupied program storage space will be displayed.

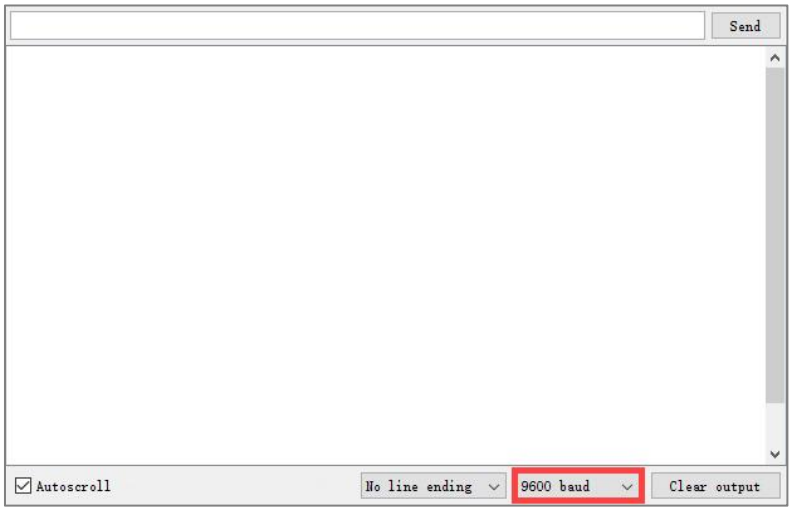


9) After compiling, click on  icon to upload the program to the development board. The status area will display "Compiling->Uploading->Complete" in turn. After uploading, the status area will stop printing the uploading information.



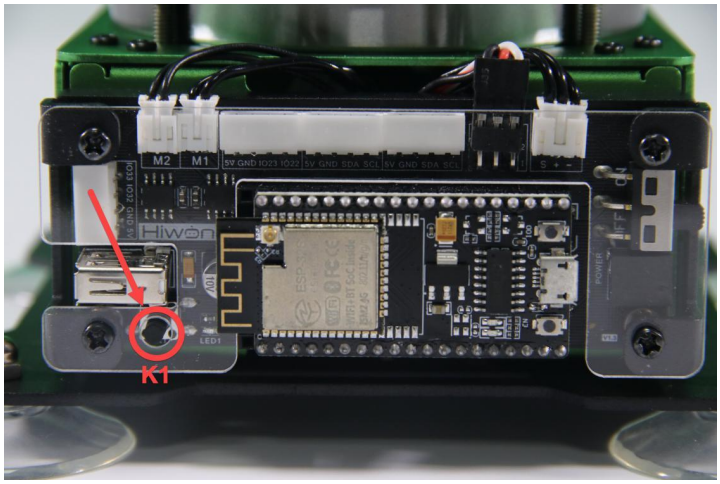
10) Then click on the serial monitor icon  in the upper right corner.

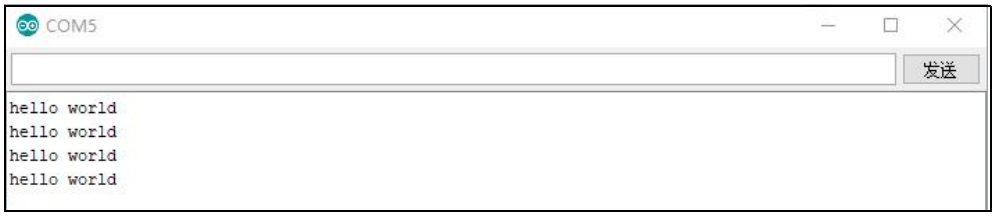
11) Select “115200 baud rate” in the pop-up window.



4. Project Outcome

When pressing K1, the serial monitor window will print “hello world”.


M2
M1
K1


COM5
发送
hello world
hello world
hello world
hello world