Mapping System

2024 年 7 月 11 日

目录

1 Function Description **2**

2 list of Components **2**

3 Pin Connection List **2**

4 Program Description **3**

4.1 ”Communication Between Computer and STM32 Microcontroller - LabVIEW Program 4

4.1.1 Front Panel 4

4.1.2 LabVIEW Block Diagram 7

4.2 STM32 Program 10

5 Usage Steps **11**

6 Warning&Read before use **12**

1 Function Description

The galvanometer system is used for optical experiments requiring microfield scanning (such as local transmission spectrum measurements). It can achieve microfield scanning within an area of 20-30 μm, with a minimum step size for laser spot movement of approximately 0.05-0.06 μm (closer to 0.053 μm). The galvanometer motor has no significant backlash.

Scanning Shape: a rectangular area with length and width on the scale of tens of micrometers.

2 list of Components

1. STM32F103C8T6(link)
2. STLINK (link)
3. Galvanometer and Driver Board (link)
4. DAC module(DAC8563) (link)
5. USB to Serial Module (link)
6. Connecting Cables
   (a) 2m female to female thick Dupont wires (4 pieces) (STM32 ⟷ USB to serial module, DAC ⟷ STM32)
   (b) 20cm female to female thick Dupont wires (approximately 11 pieces) (STM32 ⟷ DAC)
   (c) Female Dupont wire to XH2.54 (2 pieces) (Galvanometer driver board ⟷ DAC)
   (d) One type of unspecified connector (included with galvanometer purchase) to bare wires (2 pieces) (Galvanometer driver board ⟷ power supply)
   (e) Cables between galvanometer driver board and galvanometers (2 pieces) (included with galvanometer purchase)

3 Pin Connection List

1. STLINK⟷STM32
   (a) SWCLK⟷DCLK
   (b) SWDIO⟷DIO
   (c) GND⟷GND
   (d) 3.3V⟷3.3V

2. USB to Serial Module⟷STM32
    (a) TXD⟷A3
    (b) RXD⟷A2
    (c) GND⟷GND(STM32)
3. DAC⟷STM32
    (a) CLR⟷A4
    (b) DIN⟷A5
    (c) SYNC⟷A6
    (d) VCC⟷3.3V(STM32)
    (e) SCLK⟷B0
    (f) LDAC⟷B1
    (g) GND⟷GND(STM32)
4. DAC⟷Driver Board
    (a) V1⟷Positive Input Terminal (+) of Driver Board 1
    (b) GND⟷Negative Input Terminal (-) of Driver Board 1
    (c) V2⟷Positive Input Terminal (+) of Driver Board 2
    (d) GND⟷Negative Input Terminal (-) of Driver Board 2

图 1: Signal Input Terminal of Driver Board

Note: For the signal input terminals of the driver board as shown in the diagram, only connect the + and - signals (these voltages drive the galvanometer motors); D does not need to be connected.

4 Program Description

Main Idea: The computer acts as a host machine, transmitting information via the USART serial port (using LabVIEW) to the STM32 microcontroller. The STM32 microcontroller controls the DAC module to output voltage to the galvanometer driver board based on received signals. By adjusting the voltage, the angle of deflection of the galvanometer mirrors is controlled.

4.1 "Communication Between Computer and STM32 Microcontroller - LabVIEW Program

This section mainly involves programming with LabVIEW. Detailed program code can be found here.

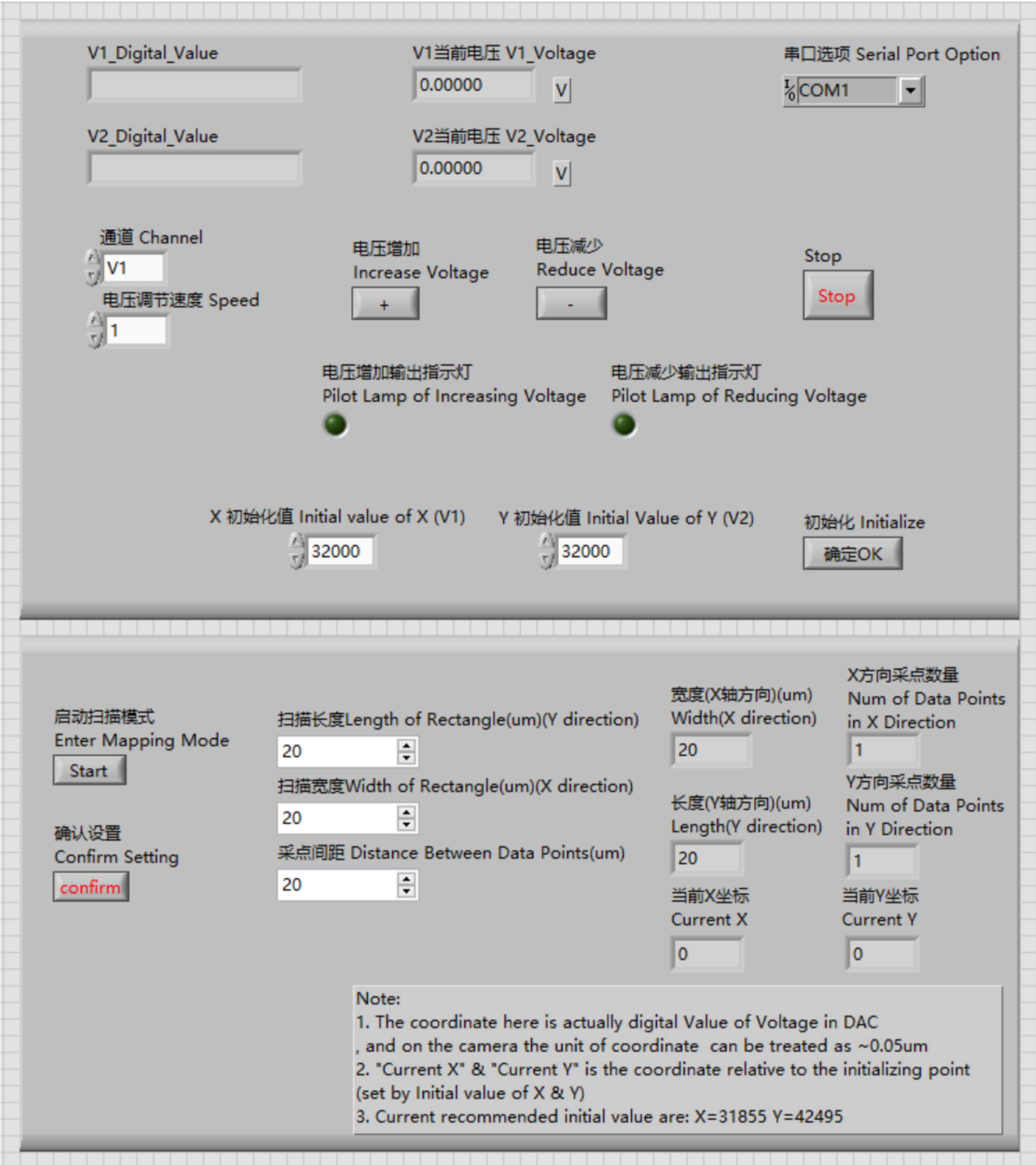


图 2: LabVIEW front panel

4.1.1 Front Panel

The front panel is mainly divided into two parts

1. Voltage Adjustment and Initialization Section

   (a) The range of "V1_Digital" and "V2_Digital" is 0~65535 (using 16-bit precision DAC), which are the digital set values for voltage output in the DAC module, corresponding to analog output voltages from -10V to +10V. ”V1_Digital” and ”V2_Digital” represent real-time feedback from the STM32 microcontroller after receiving signals from the computer. These digital values of voltage for the two channels can be interpreted as XY coordinates of the laser spot.

(b) ”V1 Current Voltage” and ”V2 Current Voltage” display the analog results after converting DAC output voltages from digital values.

(c) ”Serial Port Options” is used to select the serial port number for communication between the computer and STM32 microcontroller.

(d) ”Channel” adjusts whether the current control is for V1 or V2 (corresponding to the XY deflections of two galvanometer mirrors). **Note: In this program’s logic, V1 is always associated with X-axis offset and V2 with Y-axis offset.**

(e) There are three levels of voltage adjustment speed, corresponding to changes in digital values by 1/10/100 each time the increase or decrease voltage buttons are triggered (voltage adjustment speed 3 is the fastest). The button action can be configured, such as triggering only when pressed or continuous triggering until released.

(f) The initialization button sets the position of the laser spot to the values entered in the first two initialization boxes. **Note: The XY initialization values are also used as the center of the rectangular scanning area, so ensure correct initialization values are set before scanning.** Typically, for an established optical path, the **best initialization values are fixed, representing the laser’s normal incidence point** to minimize deformation of the laser spot during scanning. The current ideal initialization values are recorded in the front panel ”Note” third line.

2. Mapping Section

The core logic of the Mapping section is to use the current XY (V1 & V2) initialization values as the origin. Users then input the length and width of the scanning rectangle, as well as the sampling interval. Subsequently, it automatically performs a zigzag scan pattern measurement.

(a) After clicking on the ”Enter Mapping Mode” button, the system enters scanning mode. At this point, a prompt box appears, asking the user to enter the scan width, scan length, and sampling interval.

图 3: Prompt Box 1

(b) After entering the necessary scan information, the information box on the right will display details such as the number of sampling points in the X direction, allowing users to assess whether the number of points is appropriate and whether the measurement time is too long.

(c) After entering the length, width, and interval in the input fields on the front panel, and then clicking the "Confirm Setting" button, the system will prompt the outline of the scanning area to be displayed.

图 4: Prompt Box 2

Actually, at this point, the system will not outline the entire contour. Instead, it will **sequentially move to the four vertices of the rectangle** to allow users to confirm the correct position of the scanning area.

(d) After scanning the contour, the system will pop up a final prompt box asking the user whether they want to scan the selected area.

图 5: Prompt Box 3

If the user clicks "Confirm," the system will proceed with the scanning operation. If "Cancel" is chosen, the system will exit the scanning mode.

4.1.2 LabVIEW Block Diagram

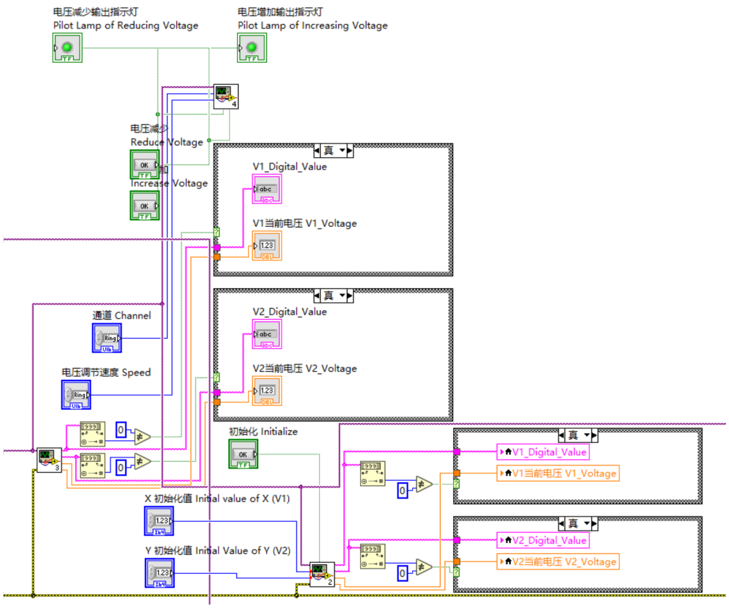


图 6: Block Diagram1

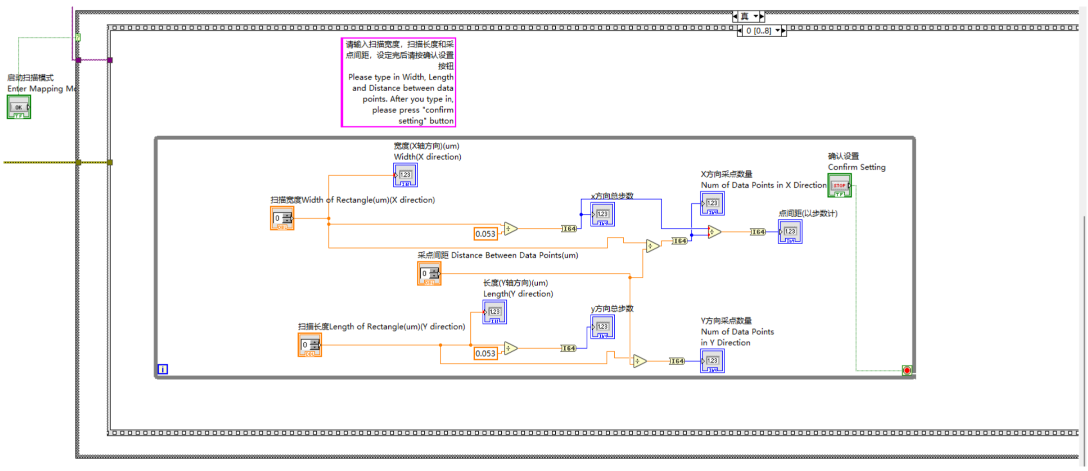


图 7: Block Diagram2

The LabVIEW program block diagram also consists of two parts: Voltage Adjustment/Initialization corresponds to 'Block Diagram 1', and Scanning Operations corresponds to 'Block Diagram 2'.

The three core subVIs are

1. Voltage Adjustment SubVI

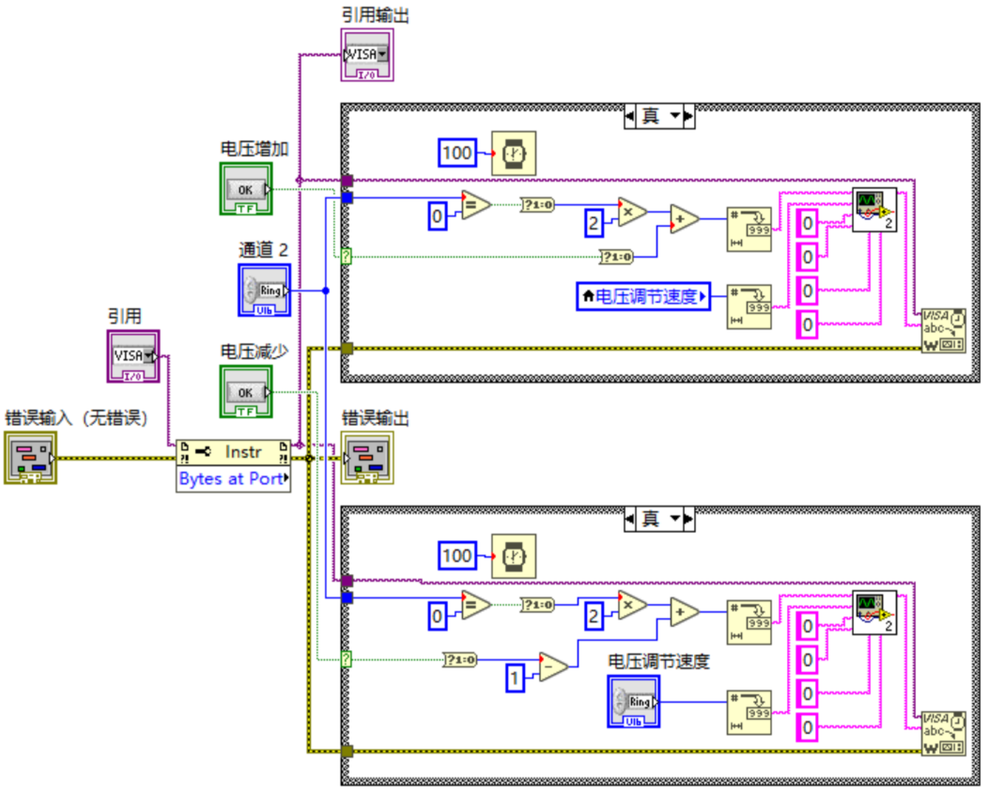


图 8: Voltage Adjustment SubVI

This subVI corresponds to manually adjusting the voltage values of V1 and V2. After selecting the 'Channel,' pressing the voltage increase/decrease buttons causes the computer to send a data packet of 6 valid data bytes to the STM32 via the serial port. The data packet is actually eight bits long, starting with the character 'F' and ending with the character 'E', marking the beginning and end of the data packet. The middle six bits are defined as

(a) First position: '3' corresponds to increase V1, '2' corresponds to decrease V1, '1' corresponds to increase V2, '0' corresponds to decrease V2.

(b) Second position: '0' corresponds to voltage adjustment speed 1, '1' corresponds to voltage adjustment speed 2, '2' corresponds to voltage adjustment speed 3.

The remaining bits are all 0.

Example: If the computer sends a string (or data packet) "F300000E" to STM32, then STM32 will increase the digital value of V1's voltage by 1.

2. Voltage Reading SubVI

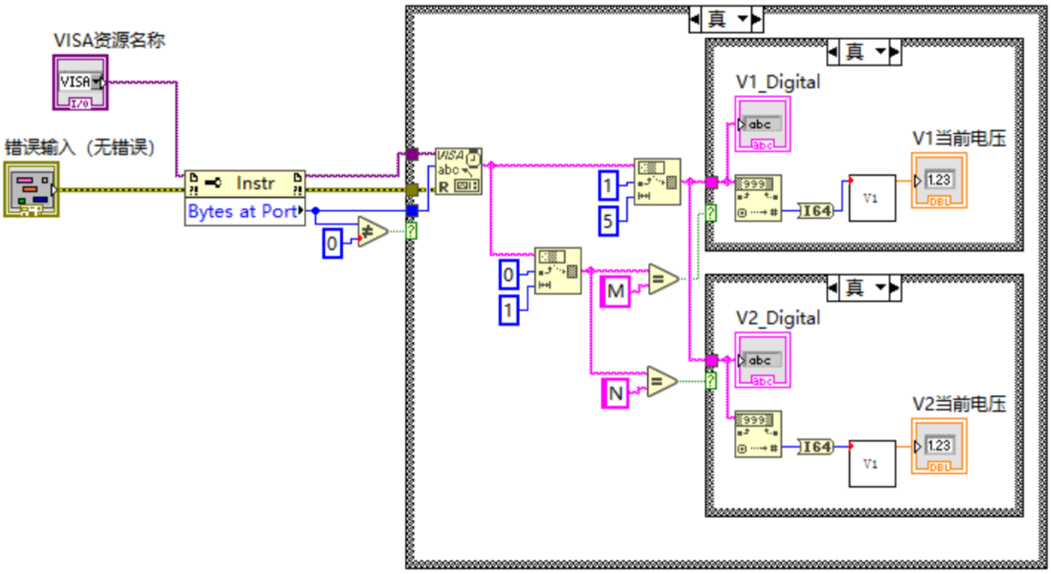


图 9: Voltage Reading SubVI

In the STM32 microcontroller, once the voltages V1 and V2 change, they are sent back to the computer as the digital voltage values of two channels. Once the computer receives information from STM32, the output of 'Bytes at Port' becomes non-zero, and the serial port reading function is then executed. The information received by the computer about the current V1 and V2 voltage values from STM32 is a string of length 6. This format has been specified in the STM32 code.

(a) If the first character of the string is 'M', then the received value is the Digital value of V1.

(b) If the first character of the string is 'N', then the received value is the Digital value of V2.

3. Voltage Setting and Reading SubVI

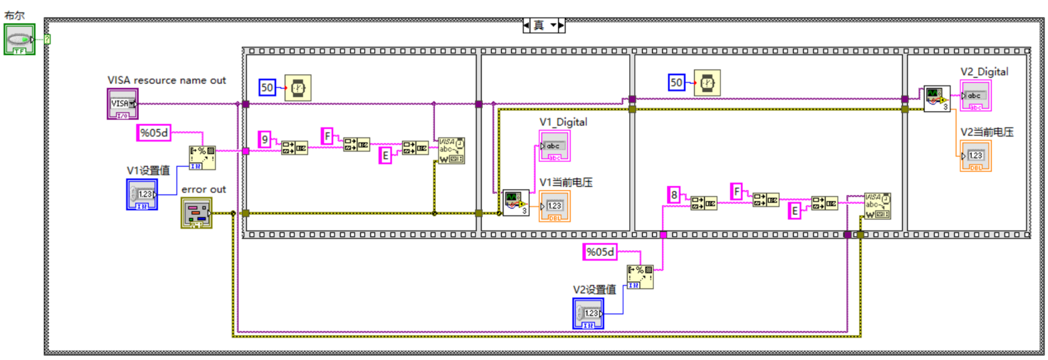


图 10: Voltage Setting and Reading SubVI

The method by which this SubVI sets the voltage is still through sending a data packet via the serial port to the STM32, with the same format as described in the Voltage Adjustment SubVI. However, the protocol differs.

(a) If the first character of the data packet's valid data is '9', it corresponds to directly setting the Digital value of V1. The subsequent five characters represent the setting value of V1_Digital.

(b) If the first character of the data packet's valid data is '8', it corresponds to directly setting the Digital value of V2. The subsequent five characters represent the setting value of V2_Digital.

Here are the two points to note in the design of this SubVI:

(a) If the Digital setting value in the data packet is less than five digits, for example, 3000, it should be padded with zeros at the front using to ensure the length of the setting value is 5. ("3000" → "03000")

(b) The reason for combining voltage setting and reading into the same SubVI is to ensure that feedback information received from STM32 after setting the first channel's Digital value is not overwritten by feedback when setting the second channel. Therefore, a sequential structure is used here: after setting the Digital value of the first channel, feedback information is immediately read, then the Digital value of the second channel is set, followed by reading the feedback information again.

Once the basic functionalities of setting and reading are in place, implementing the scanning operation is achievable by incorporating appropriate event and loop structures.

4.2 STM32 Program

The program in STM32 is primarily written in C language and can be divided into three main parts: UART hardware driver (using USART communication protocol), DAC module hardware driver (using SPI communication protocol), and the main program. The hardware driver programs mainly include initialization and basic read-write operations. The main program defines the meanings of data packets and includes conditional statements to change the Digital values of voltages based on certain conditions.

STM32's program is written using Keil uVision5 . Once the program is completed, you can compile and download it to the STM32 microcontroller by clicking on the compile and download icons in sequence within the program.

Detailed program code can be found here.

5 Usage Steps

1. connecting devices

   Using the pin connection list as reference, connect the following parts: galvanometer driver board to galvanometer, galvanometer driver board to DAC module, DAC module to STM32 microcontroller, STM32 microcontroller to USB to serial module, and STM32 to ST-LINK.

   Note: Due to the program logic of the galvanometer system where V1 corresponds to the X-axis and V2 to the Y-axis, ensure that the connections between the DAC module and the galvanometer driver board are consistent with this program logic.

2. Connect the power supply and the galvanometer driver board

   Note: The power supply can use the Zhaoxin dual-channel DC power supply.

图 11: Zhaoxin dual-channel DC power supply

   When powering on, adjust the power supply to series mode, with each side set to 24V (ensure sufficient current is provided to achieve the set voltage). In this configuration, the power supply operates in a positive and negative series DC mode. The power supply manual specifies which interface serves as the positive pole, negative pole, and ground (GND) in this mode, as indicated in the diagram. Connect the positive (+), negative (-), and GND terminals of the power supply to the corresponding positive (+), negative (-), and GND terminals on the driver board.

3. Connect STLINK (STLINK is used to download microcontroller programs and power microcontroller) and the serial port to the computer, open the LabVIEW program, and select the corresponding serial port number.

4. Run the LabVIEW program, select the channel and adjust the speed, then click the button to adjust V1 and V2.

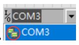
5. If you want to do Mapping, first select a suitable initialization point (origin). (This point is determined by the optical path. If the optical path is determined, then the best origin is also determined, that is, the positive incident point of the laser. This is done to prevent the laser spot from being significantly deformed during the scanning process. **This point is recorded in the third item of "Note" on the front panel, and can be updated and adjusted according to the actual situation.**。)Enter the initial values for XY and click the Initialize button. (This initial point will be the center of the scan area.)

6. Click the "Enter Mapping Mode" button, the system will prompt you to enter the length, width and sampling point spacing of the scanning area, and then the user can enter the length, width and sampling point spacing of the scanning area.

7. Click the "Confirm Settings" button, and a pop-up window will pop up to remind you that the outline of the scan area will be displayed next. After clicking Confirm, the system will move to the four vertices of the scan rectangle so that the user can confirm whether the position of the scan area is correct.

8. After the system has finished scanning the outline, a final prompt box will pop up, asking the user if they want to scan the area. If the user clicks OK, the system will perform the scan. If the user chooses Cancel, the system will exit the scan mode.

9. After the scan is completed, the system will automatically exit the scanning mode.

Note: When terminating the program，**You need to press the STOP button**，Note that you .**cannot terminate execution using the Terminate Execution button in LabVIEW**，This will cause the serial communication session to not be closed properly, which is manifested by the red circle icon still remaining in the serial port list after the program is terminated.，This means that the serial port is being used. If you have not unplugged the serial port, you can run the labVIEW program again and click Stop to close the serial communication session. If you have unplugged the serial port, you need to completely close the labVIEW program and reopen it. If the occupied icon disappears, it means that the port status is normal.

6 Warning & Read before use

When using, be sure to confirm that **do not place electronic devices such as the STM32 driver board directly on the optical platform to avoid short circuit damage to the device**. It must be placed in an insulated place when in use.