SDK Calibration Demo Description Document

Check whether the device is powered and properly connected before calibration. Note that the USB interface should be connected to the USB3.0 interface of the computer. The following is the operating instructions of the SDK calibration demo. The process is shown in figure 1.

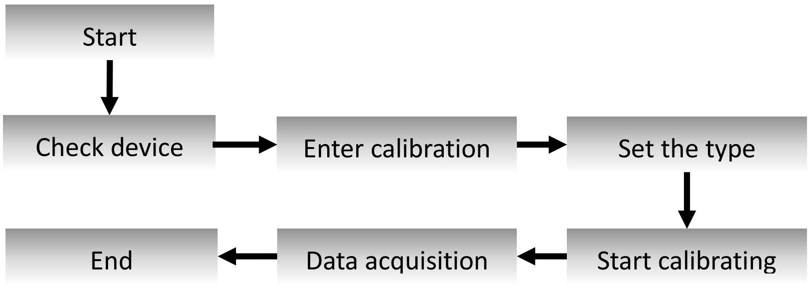


Figure 1 Process of SDK Calibration Demo

The following conditions require the scanner to calibrate:

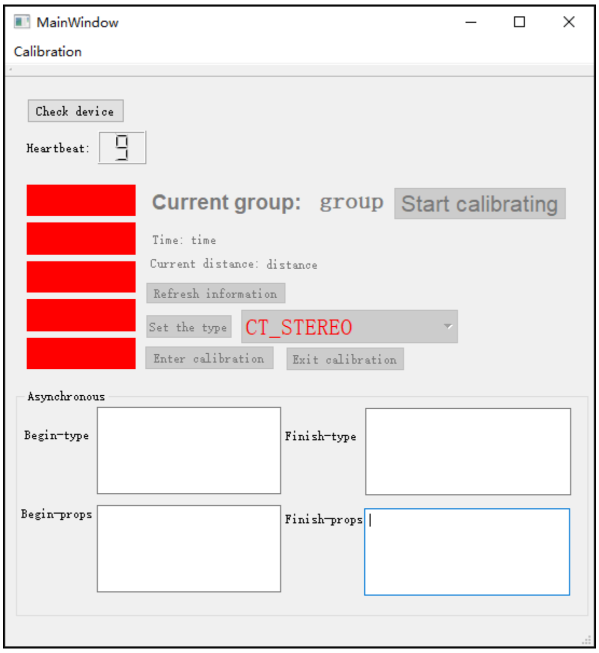
- Initial use of scanners;
- Use after a long time;
- Scanning data is incomplete and data quality is seriously degraded during scanning.

Should pay attention to the calibration process:

- Make sure the calibration plate is clean and free of scratches;
- Make sure to use the calibration plate corresponding to the device for calibration.

1、Start

Open the SDK Calibration, the interface is shown in figure 2.



Heartbeat: 9

Current group: group

Time: time

Current distance: distance

Set the type: CT_STEREO

Figure 2 SDK Calibration Interface

2、Check Device

As shown in figure 3, click the "Check device" button to enter the detection status of the device, and a progress bar indicates the progress of the detection.

Corresponding functions: mainwindow.cpp-> on_pushButton_DeviceCheck_clicked

“Heartbeat” text: Server heartbeat(When the value is 0, the delegate server dies and can no longer provide data to the client). mianwindow.cpp->onHeartbeat

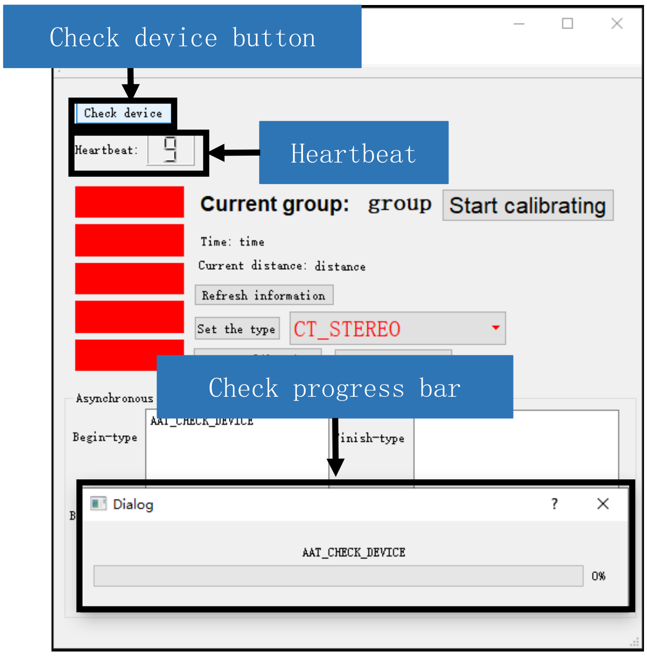


Figure 3 "Check device" button interface diagram

3、Enter calibration

Click the "Enter calibration" button to enter the calibration state, and the interface will display a progress bar, as shown in figure 4.

Corresponding functions: mainwindow.cpp-> on_pushButton_enterCali_clicked

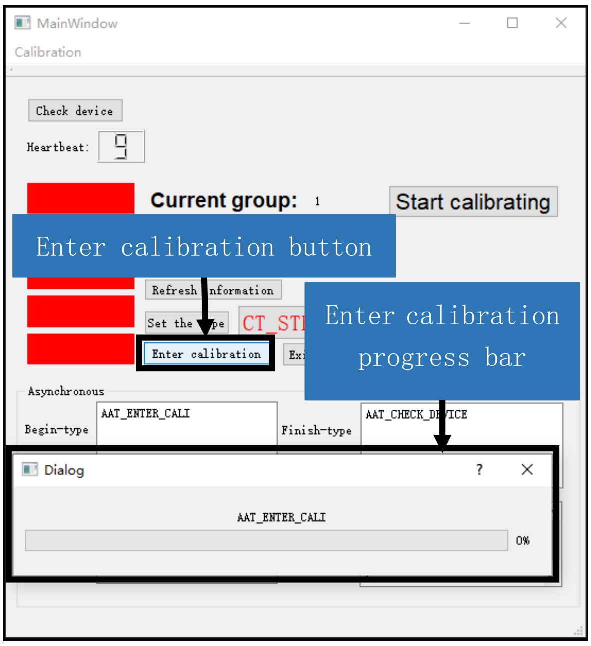


Figure 4 Enter the calibration state diagram

4、Set the type

Select the calibration type according to the need. As shown in figure 5, the details of the calibration type parameters are shown in table 1.

Corresponding functions: mainwindow.cpp-> mainwindow.cpp-> on_pushButton_CaliSetType_clicked

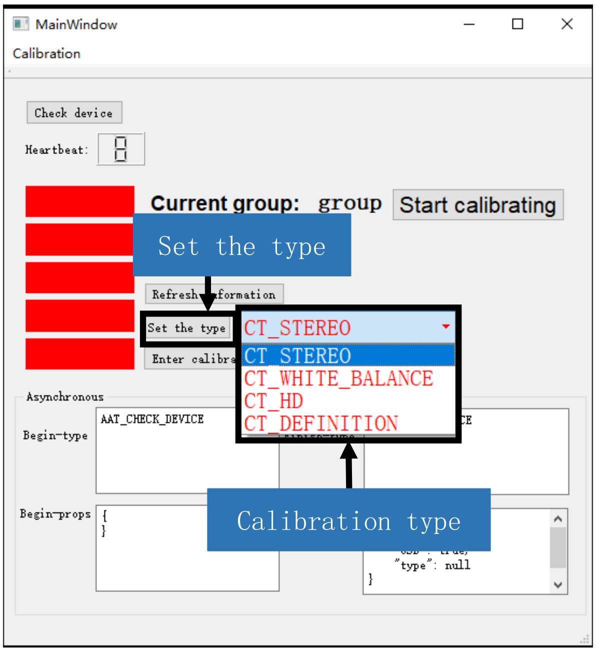


Figure 5 Set the calibration type diagram

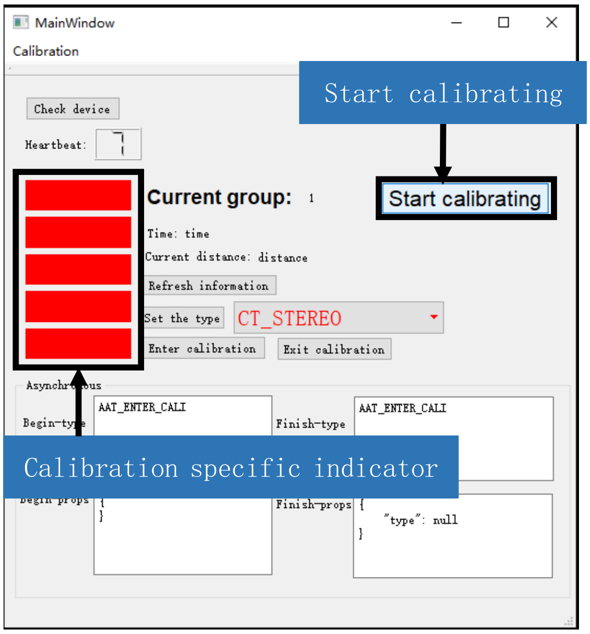
Table 1 Calibration type parameters

Parameter	Implication	UI Type	Function
Set the type	Set the calibration type: CT_STEREO CT_WHITE_BALANCE CT_HD CT_DEFINITION	Button+ Drop down list	mainwindow.cpp-> on_pushButton_CaliSetType_clicked

5、Start calibrating

The scanner calibration requires five position, five images for each position, and each position scanner should be perpendicular to the horizontal plane. The five positions of the calibration plate are placed horizontally, and the calibration plate is placed on the bracket to fix different sides. The click the “start calibration” button to start the calibration, as shown in figure 6.

Corresponding functions: mainwindow.cpp-> on_pushButton_SetSnapEnabled_clicked



Current group: 1

Time: time

Current distance: distance

Set the type: CT_STEREO

Begin-type: AAT_ENTER_CALI

Finish-type: AAT_ENTER_CALI

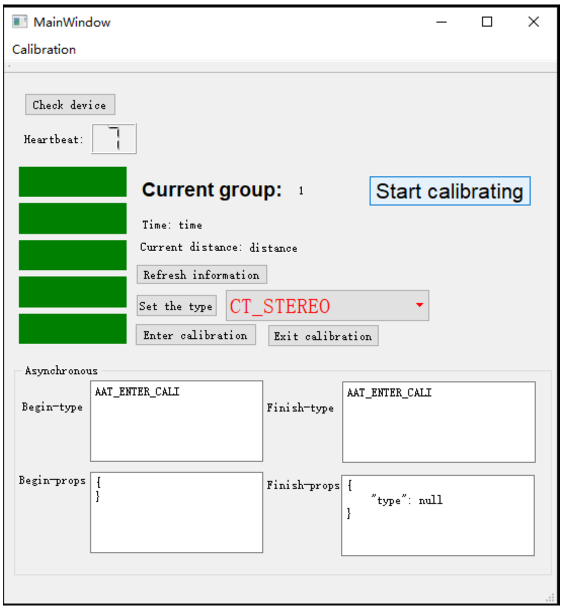
Begin-props: { }

Finish-props: { "type": null }

Figure 6 Start calibration diagram

To start the calibration, the distance between the scanner and the calibration plate needs to be adjusted. The five indicator bars in the calibration distance frame in figure 6 represent the distance between the scanner and the calibration plate from bottom to top, respectively 350mm, 375mm, 400mm, 425mm, 450mm.

Each group needs to collect different photos of these 5 positions. During the collection process, the LED lights flash, moving the scanner from top to bottom or from bottom to top until the distance bar is filed with green. As shown in figure 7, location image acquisition is complete. In each set of calibration process, the scanner cross or grids should not deviate from white square area in the center of the calibration plate, and the calibration plate should not be moved.



Current group: 1

Time: time

Current distance: distance

Set the type: CT_STEREO

Begin-type: AAT_ENTER_CALI

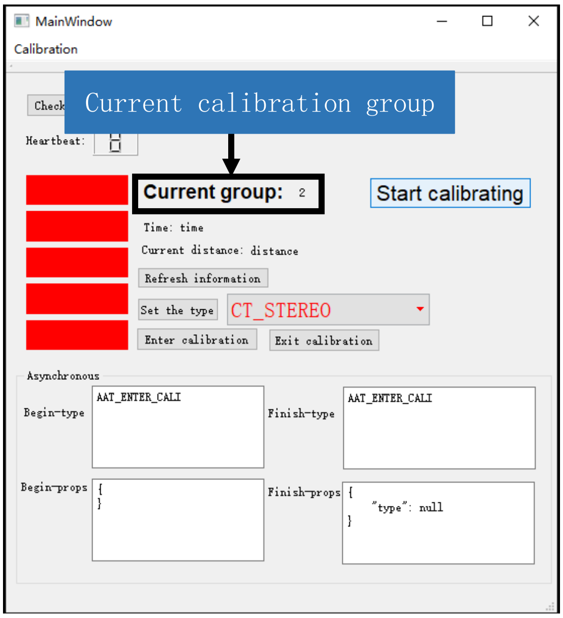
Finish-type: AAT_ENTER_CALI

Begin-props: { }

Finish-props: { "type": null }

Figure 7 A set of calibration completion sketches

After a set of data acquisition is completed, it will automatically jump to the next set of calibrations. The interface is shown in figure 8. Adjust the position of the calibration board and click the “Start calibrating” button to start the calibration of current group.



Current group: 2

Time: time

Current distance: distance

Set the type: CT_STEREO

Begin-type: AAT_ENTER_CALI

Finish-type: AAT_ENTER_CALI

Begin-props:

```
{
}
```

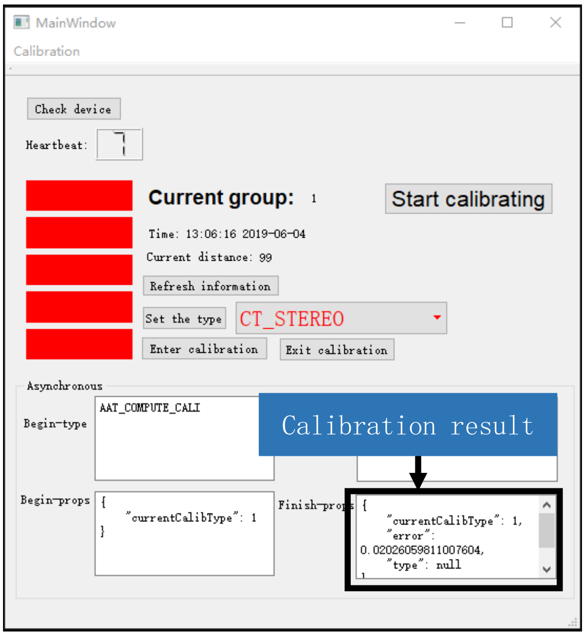
Finish-props:

```
{
    "type": null
}
```

Figure 8 Enter the next set of calibration sketches

6、 End calibration

When the calibration board is completed at five different locations, it will automatically enter the data processing process and be represented by a progress bar. The data processing results are displayed in the Finish-props dialog box, as shown in figure 9. Details of the parameters selected in the box in figure 10 are shown in table 2.



Current group: 1

Time: 13:06:16 2019-06-04

Current distance: 99

Set the type: CT_STEREO

Begin-type: AAT_COMPUTE_CALI

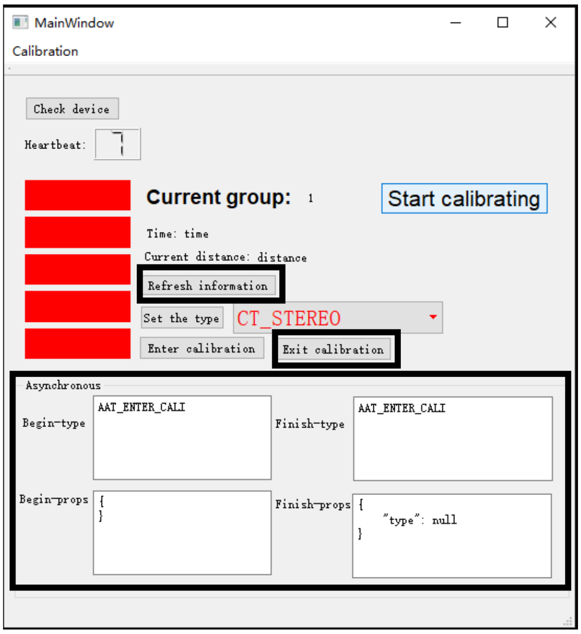
Begin-props:

```
{
    "currentCalibType": 1
}
```

Finish-props:

```
{
    "currentCalibType": 1,
    "error":
0.02026059811007604,
    "type": null
}
```

Figure 9 Calibration result diagram



Heartbeat: 7

Current group: 1

Time: time

Current distance: distance

CT_STEREO

Begin-type: AAT_ENTER_CALI

Finish-type: AAT_ENTER_CALI

Begin-props:

```
{
}
```

Finish-props:

```
{
    "type": null
}
```

Figure 10 Parameters in box selection

Table2 Parameter list

Parameter	Implication	UI Type	Function
Exit calibration	Exit calibration	Button	mainwindow.cpp->on_pushButton_CaliExit_clicked
Refresh information	Refresh information	Button	mainwindow.cpp->on_pushButton_GetInformation_clicked
Begin-type	Asynchronous callback begin type	Text	Mainwindow.cpp-> onPublishReceived
Begin-props	Asynchronous callback begin attribute	Text	mainwindow.cpp-> onPublishReceived
Finish-type	Asynchronous callback finish type	Text	mainwindow.cpp-> onPublishReceived
Finish-props	Asynchronous callback finish attribute	Text	mainwindow.cpp-> onPublishReceived

Click on “Exit” button to exit the scan in progress project, and start a new scan from step 3 of the description.