

# SDK Fix Scan Demo Description Document

Check whether the device is powered and properly connected before scanning. 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 Fix Scan Demo. The process is shown in figure 1.

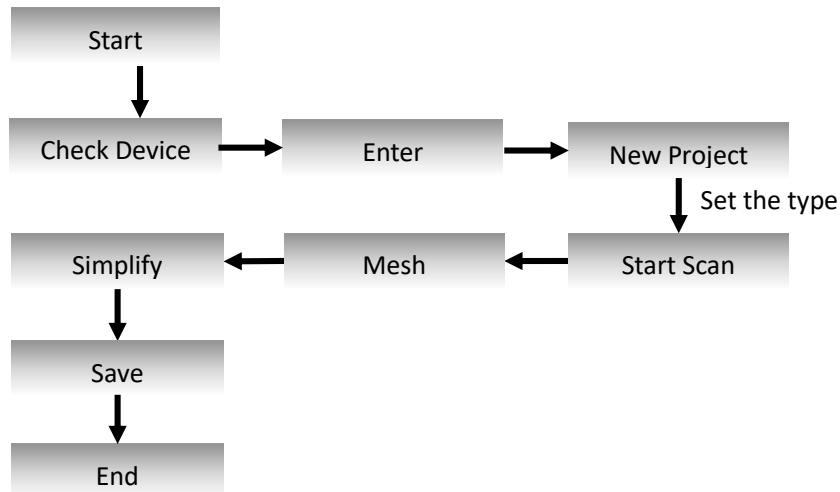


Figure 1 Process of SDK Fix Scan Demo

## 1、Start

Open the SDK Fix Scan, the interface is shown in figure 2.

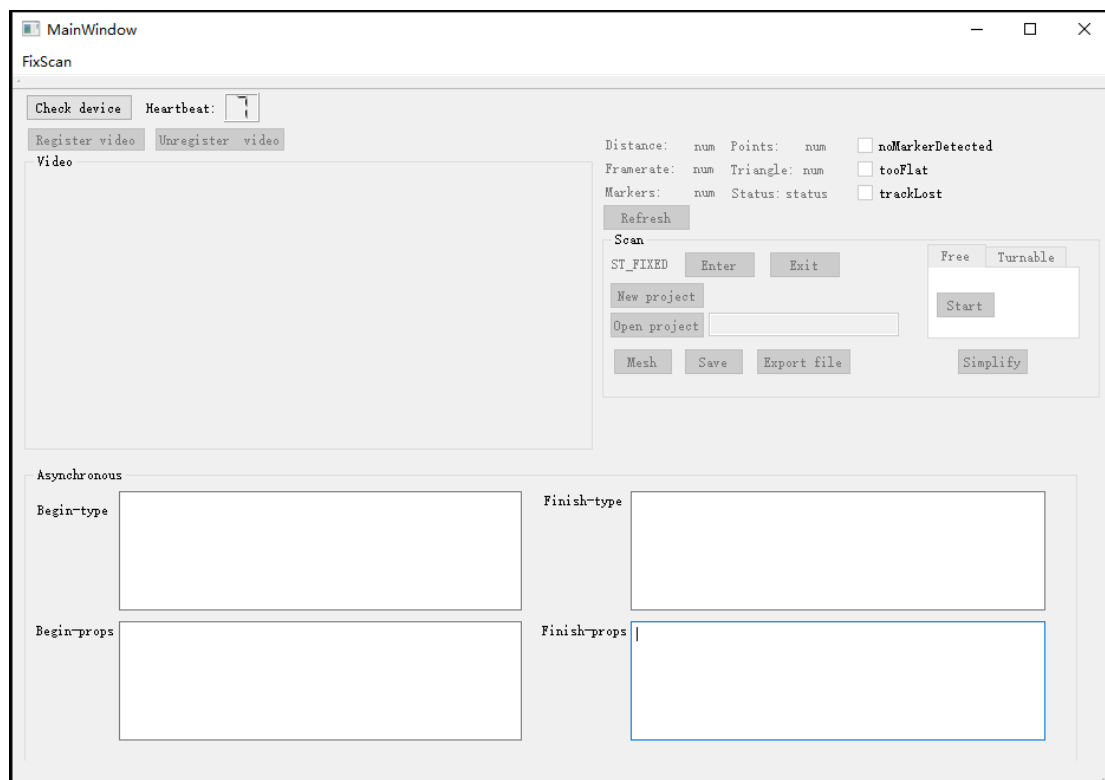


Figure 2 SDK Fix Scan Interface

## 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: `mianwindow.cpp->on_pushButton_DeviceCheck_clicked`

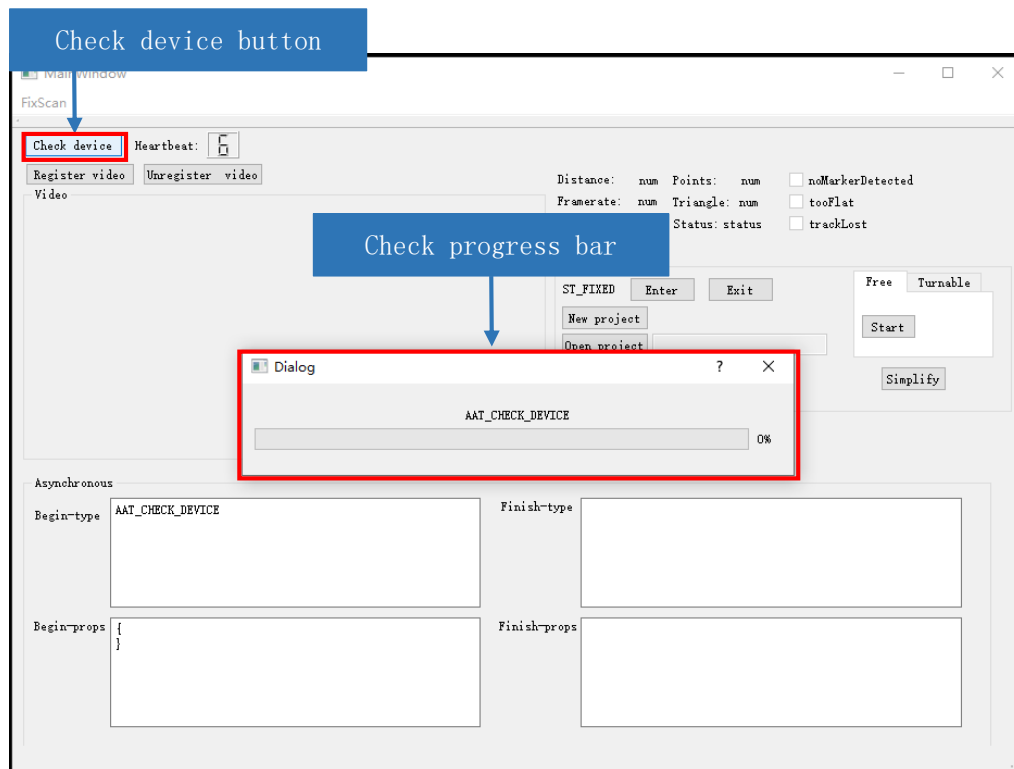


Figure 3 "Check device" button interface diagram

### 3、Enter

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

Corresponding functions: `mainwindow.cpp->on_pushButton_ScaneEnterScan_clicked`

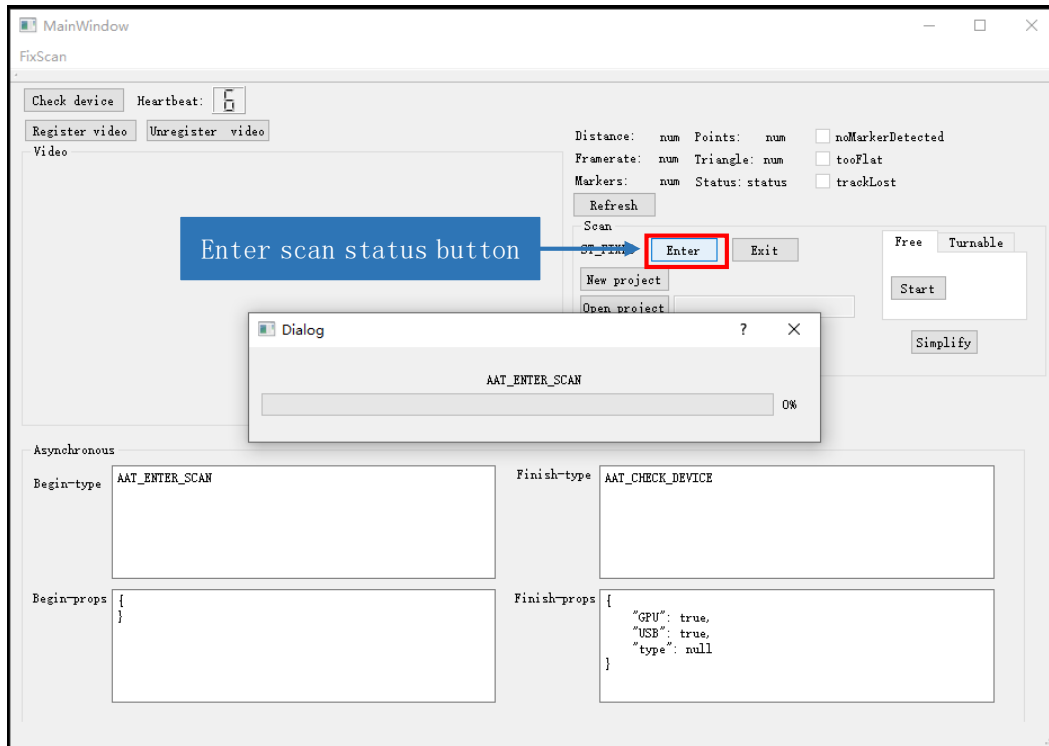


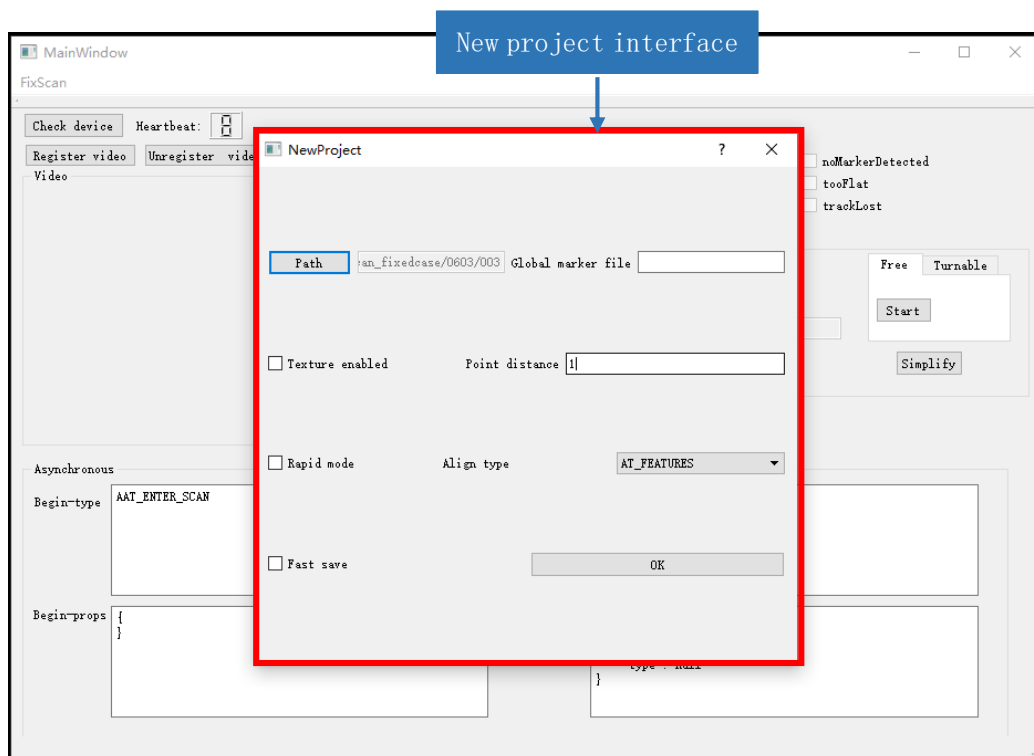
Figure 4 "Enter" button interface diagram

#### 4、New Project

Click the “New Project” button to enter the new project interface as shown in figure 5. The parameters of the new project are shown in table 1.

Corresponding functions: mainwindow.cpp->

on\_pushButton\_ScanNewProject\_clicked



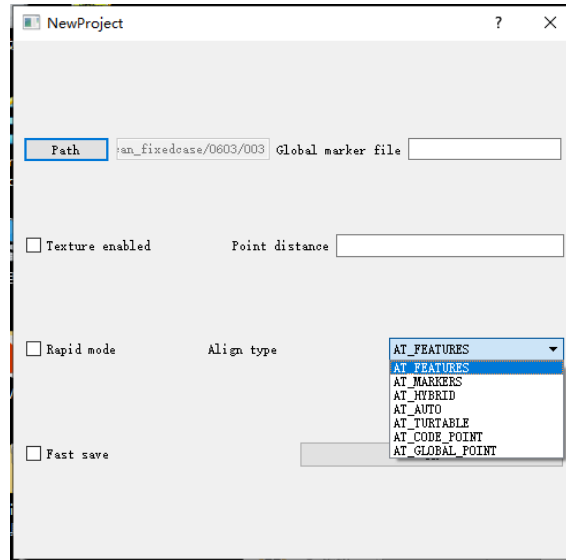


Figure 5 New Project interface diagram

Table 1 New project parameters

Parameter	Implication	UI Type	Function
Path	Select the path to save the project	Button+Text	commonui.cpp->onPathButtonClicked
Global marker file	Global frame point File Path	Button+Text	commonui.cpp->NewProject(QDialog *parent)
Texture enabled	Whether to use a texture camera	Check box	commonui.cpp->NewProject(QDialog *parent)
Rapid mode	Determine the scan frame rate	Check box	commonui.cpp->NewProject(QDialog *parent)
Fast save	Whether to save each frame of image	Check box	commonui.cpp->NewProject(QDialog *parent)
Point distance	Point distance	Text	commonui.cpp->NewProject(QDialog *parent)
Align type	Scan type: AT_FEATURES(Feature stitching) AT_MARKERS(Mark stitching) AT_HYBRID(Mixed stitching) AT_AUTO(Automatic stitching)	Drop-down list	commonui.cpp->NewProject(QDialog *parent)
OK	Confirm entry to new project	Button	commonui.cpp->onPushButtonClicked

We choose the path of the new project, set the point distance to 1, the other is the default, click the "OK" button. The interface of the new project is shown in figure 6. The parameters of the main scanner in the upper right corner are shown in table 2.

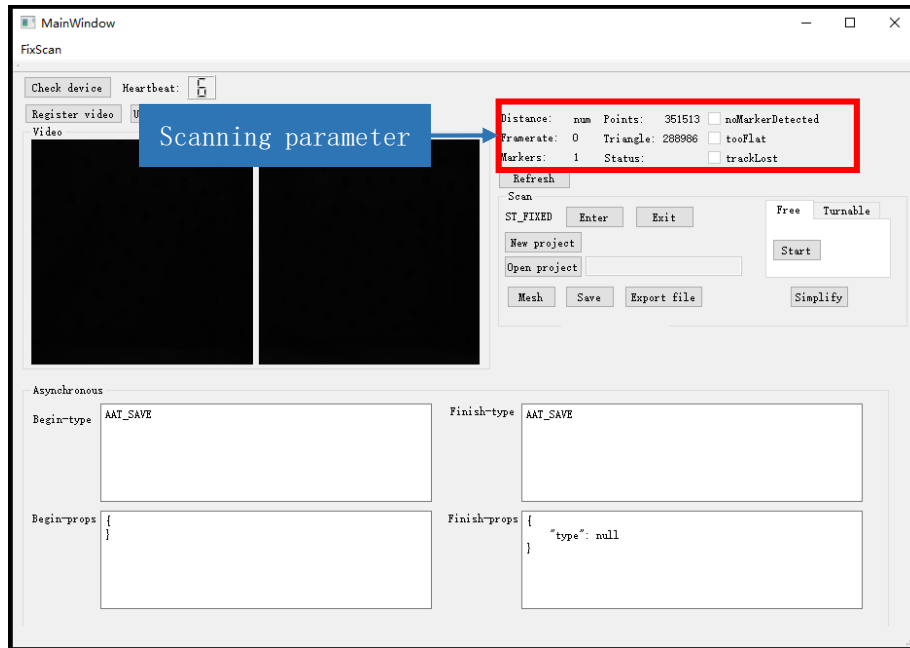


Figure 6 Scanning parameters

Table 2 Scanning parameters

Parameter	Implication	UI Type	Function
Distance	Distance between the device and the scanned object	Text	mainwindow.cpp-> ScanDist
Framerate	Scan frame rate	Text	mainwindow.cpp-> ScanFramerate
Markers	Number of point clouds scanned	Text	mainwindow.cpp-> ScanMarkerCount
Points	Model triangle face number	Text	mainwindow.cpp->ScanPointCount
Triangle	Device scan status	Text	mainwindow.cpp-> ScanTriangleCount
Status	No scan to mark point (Cannot check manually, automatically judge)	Check box	mainwindow.cpp->ScanStatus
noMarkerDetected	Too smooth (Cannot check manually, automatically judge)	Check box	mainwindow.cpp-> ScanNoMarkerDetected
TooFlat	Tracking loss (Cannot check manually, automatically judge)	Check box	mainwindow.cpp-> ScanTooFlat
trackLost	Tracking loss (Cannot check manually, automatically judge)	Check box	mainwindow.cpp-> ScanTrackLost

## 5、Start Scan

### 5.1 No turntable fixed scan (Free)

Click on the "Start" button to start scanning the model. The interface is shown in figure 7. See table 3 for details on starting the scanning interface. Here we use default parameters and click "OK" button to enter the scan .

Corresponding functions: mainwindow.cpp-> on\_pushButton\_StartFree\_clicked

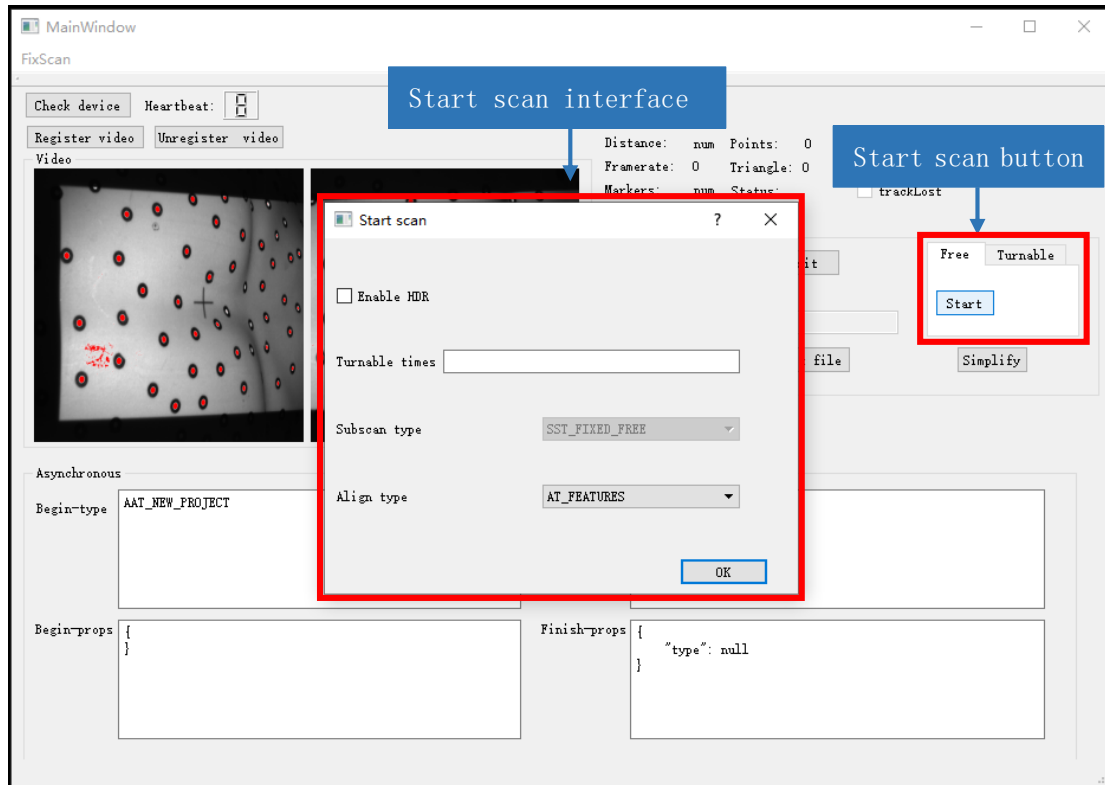


Figure 7 Start scan interface

Table 3 Start scan parameters

Parameter	Implication	UI Type	Function
Enable HDR	HDR enable	Check box	startscan.cpp->on_pushButton_clicked
Turntable times	Number of turns of the turntable	Text	startscan.cpp->on_pushButton_clicked
Subscan type	Scan subtype	Drop-down list	startscan.cpp->on_pushButton_clicked
Align type	Scan type: AT_FEATURES (Feature stitching) AT_MARKERS (Mark stitching) AT_HYBRID (Mixed stitching) AT_AUTO (Automatic stitching)	Drop-down list	startscan.cpp->on_pushButton_clicked
OK	Confirm entry to preview	Button	startscan.cpp->on_pushButton_clicked

Here our parameters are the default, click the "OK" button to enter the no turntable fixed scan mode. Click on "Register video" for camera video registration. If the previous operation is correct, the view of the scanner's left and right cameras can be displayed in real time under the video window, as shown in figure 8.

"Register video" Button: Camera Video Registration, mianwindow.cpp -> on\_pushButton\_RegisterProcessor\_clicked

"UnRegister video" Button: Camera Video UnRegistration, mianwindow.cpp -> on\_pushButton\_UnRegisterProcessor\_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

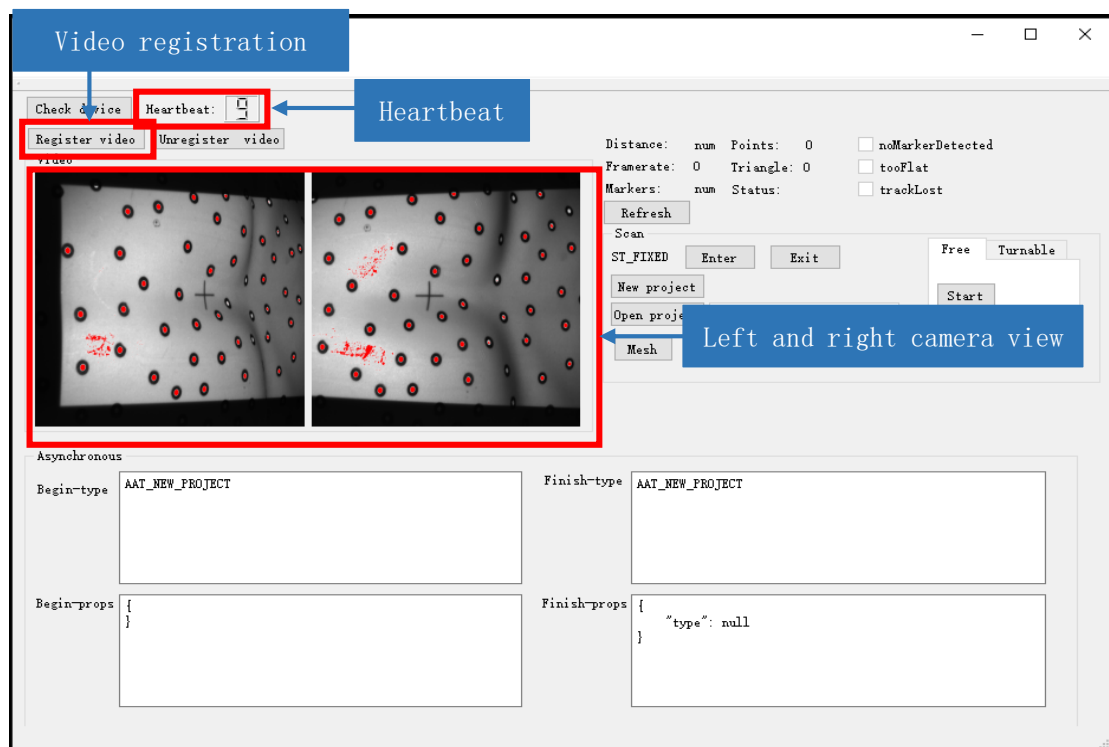


Figure 8 Video registration interface

The fixed scan without a turntable only needs to click the "Start" button, and the device automatically ends the scan after scanning one frame.

## 5.2 Turntable

In the case of fixed scanning of turntable, we choose the "Turntable" mode. As shown in figure 9, click on the "start" button to start scanning, and start scanning sub-interface parameters are consistent with those of the sub-interface when starting scanning free, details are shown in table 3. We need to set the turntable rotation times, where we set "Turntable times" to 8.

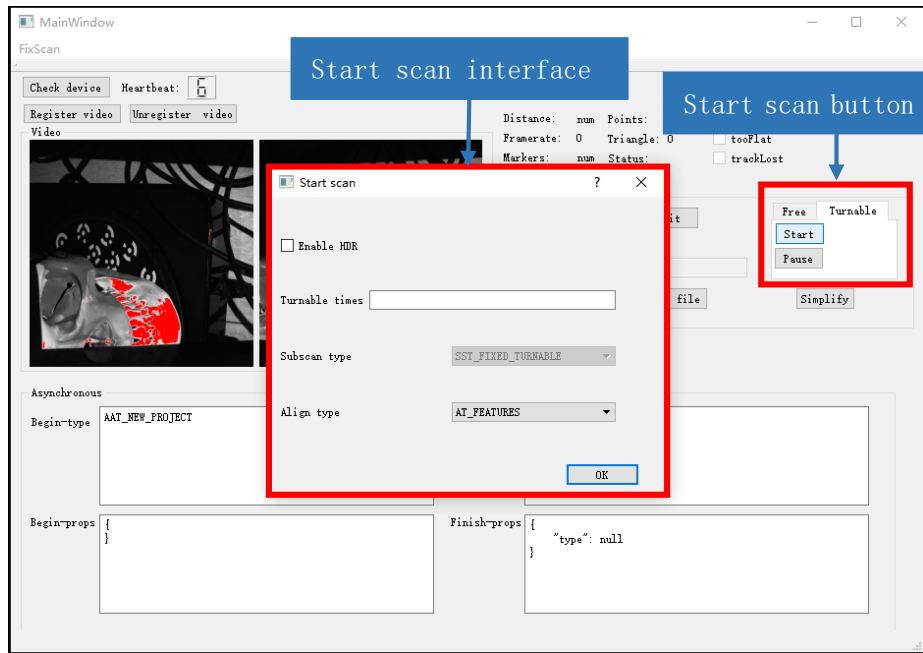


Figure 9 Start scan interface

When the “Pause” button is clicked, the scan is paused, as shown in figure 10. Pause scanning sub-interface parameters are consistent with those of the sub-interface when starting scanning free, details are shown in table 3. Stop scanning and click the “Pause” button again to start scanning again.

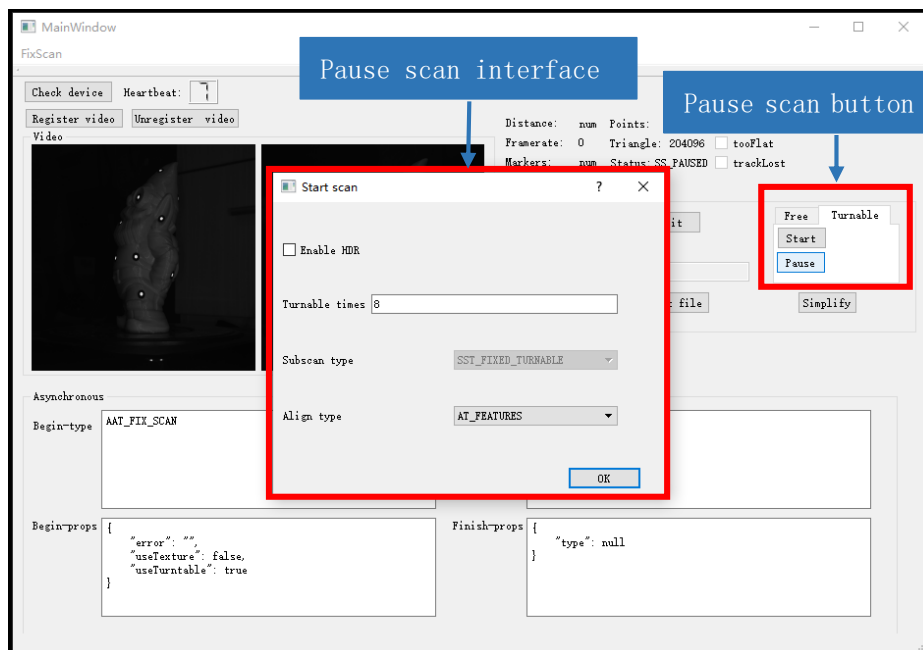


Figure 10 Start scan interface

## 6、Mesh

Click the “Mesh” button to grid the point cloud. The schematic is shown in figure 11. The parameters of the Mesh processing interface are shown in table 4. We select the default parameters for grid processing.

Corresponding functions: mainwindow.cpp-> on\_pushButton\_scanMesh\_clicked



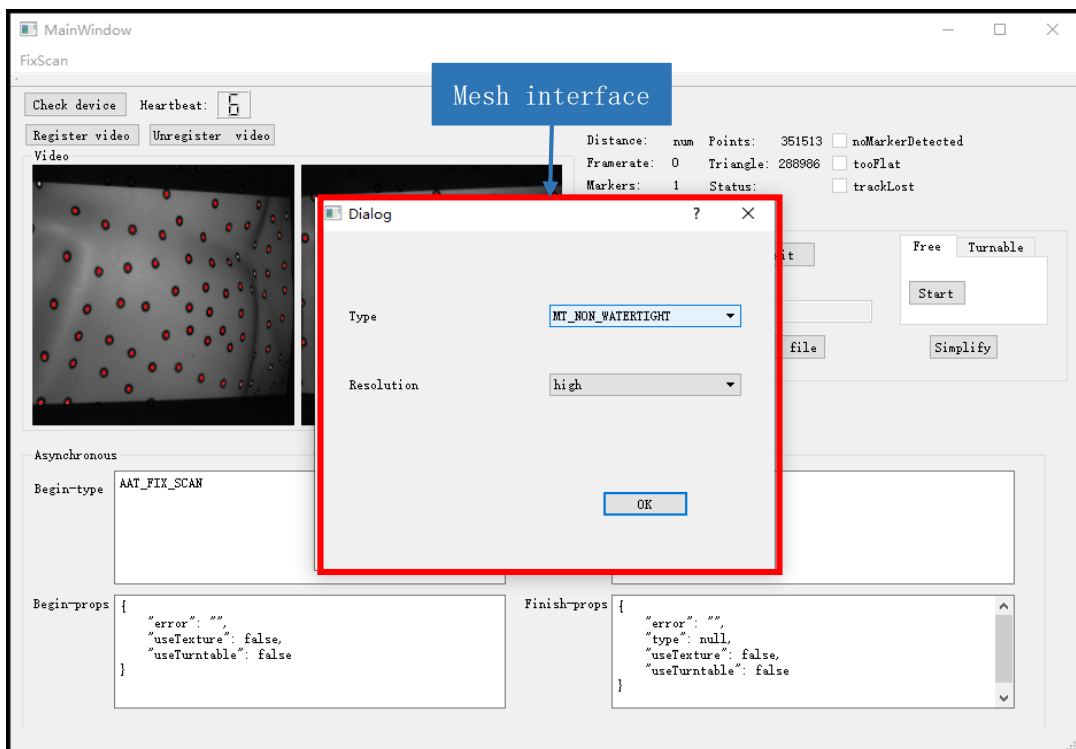
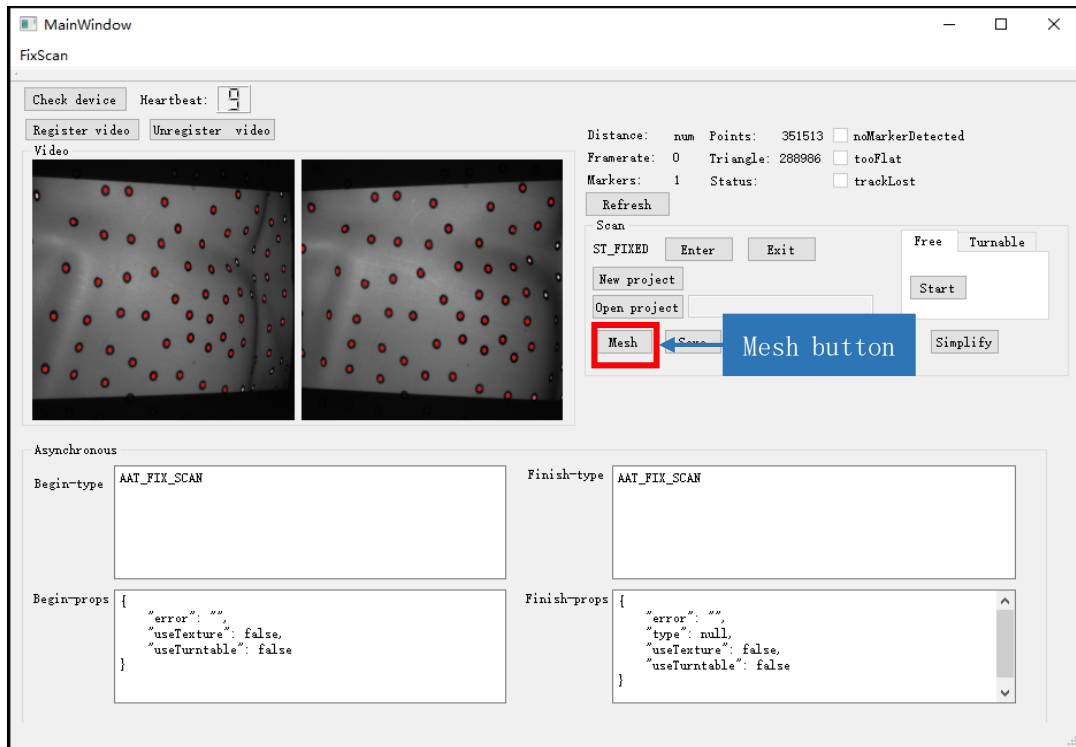


Figure 11 Mesh processing interface

Table 4 Mesh parameters

Parameter	Implication	UI Type	Function
Type	Grid encapsulation type: MT_NON_WATERTIGHT	Drop-down list	mesh.cpp-> on_pushButton_mesh_clicked

	MT_WATERTIGHT		
Resolution	Mesh density: Hight Medium Low	Drop-down list	mesh.cpp-> on_pushButton_mesh_clicked
OK	Confirm mesh process	Button	mesh.cpp-> on_pushButton_mesh_clicked

### 7、Simplify

Click the “Simplify” button to complete the simplification of the grid data. The simplified interface is shown in figure 12, the parameter details are shown in table 5. For details on the simplified parameters, check the “need mesh smooth” and “need mesh sharp” check boxes to simplify the mesh.

Corresponding functions:

mainwindow.cpp->on\_pushButton\_scanSimplify\_clicked

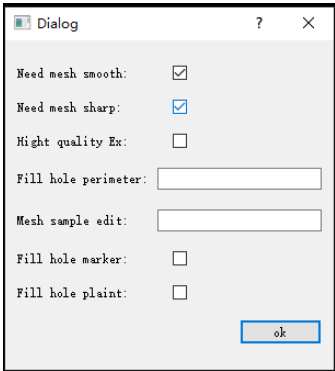
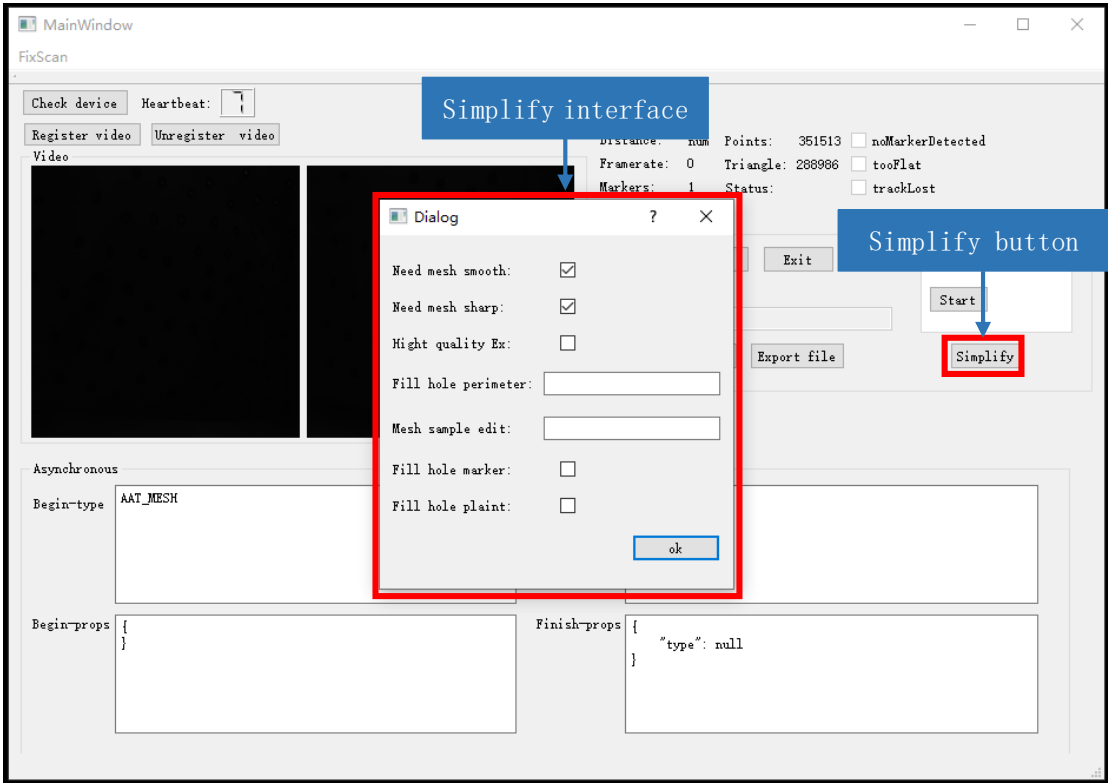


Figure 12 Simplify processing interface

Table 5 Simplify parameters

Parameter	Implication	UI Type	Function
Need mesh smooth	Mesh smoothing	Check box	simplify.cpp->on_pushButton_clicked
Need mesh sharp	Mesh sharp	Check box	simplify.cpp->on_pushButton_clicked
High quality Ex	high quality	Check box	simplify.cpp->on_pushButton_clicked
Fill hole perimeter	Fill hole parameter	Text	simplify.cpp->on_pushButton_clicked
Mesh sample edit	Simplified ratio	Text	simplify.cpp->on_pushButton_clicked
Fill hole marker	Fill hole marker	Check box	simplify.cpp->on_pushButton_clicked
Fill hole marker	Fill hole marker	Check box	simplify.cpp->on_pushButton_clicked
ok	Confirm entry into simplified operation	Button	simplify.cpp->on_pushButton_clicked

#### 8、Save

Click the “Save” button to save the model data. The interface is shown in figure 13. The details of the saved interface parameters are shown in table 6. Here we set the file save path, set the resize ratio to 100, and then click the “OK” button to save. (The size of the save scale does not affect the size of the file and will change the size of the model.)

Corresponding functions: mainWindow.cpp->on\_pushButton\_scanSave\_clicked

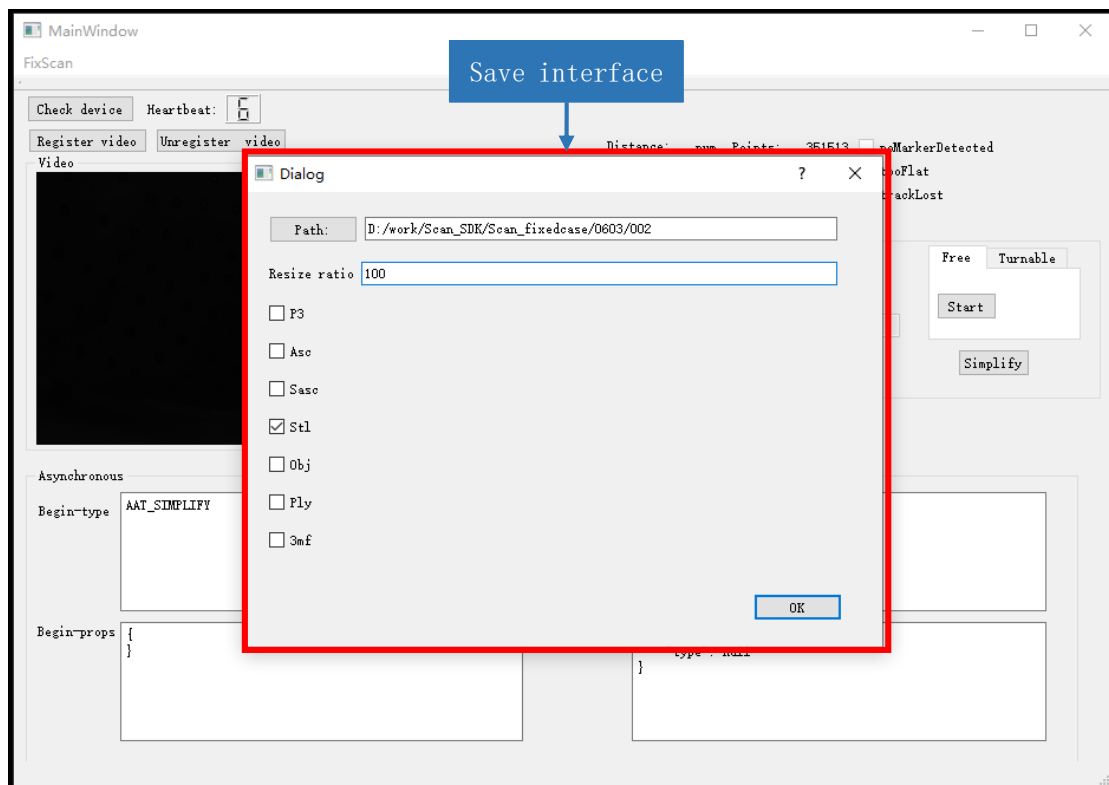
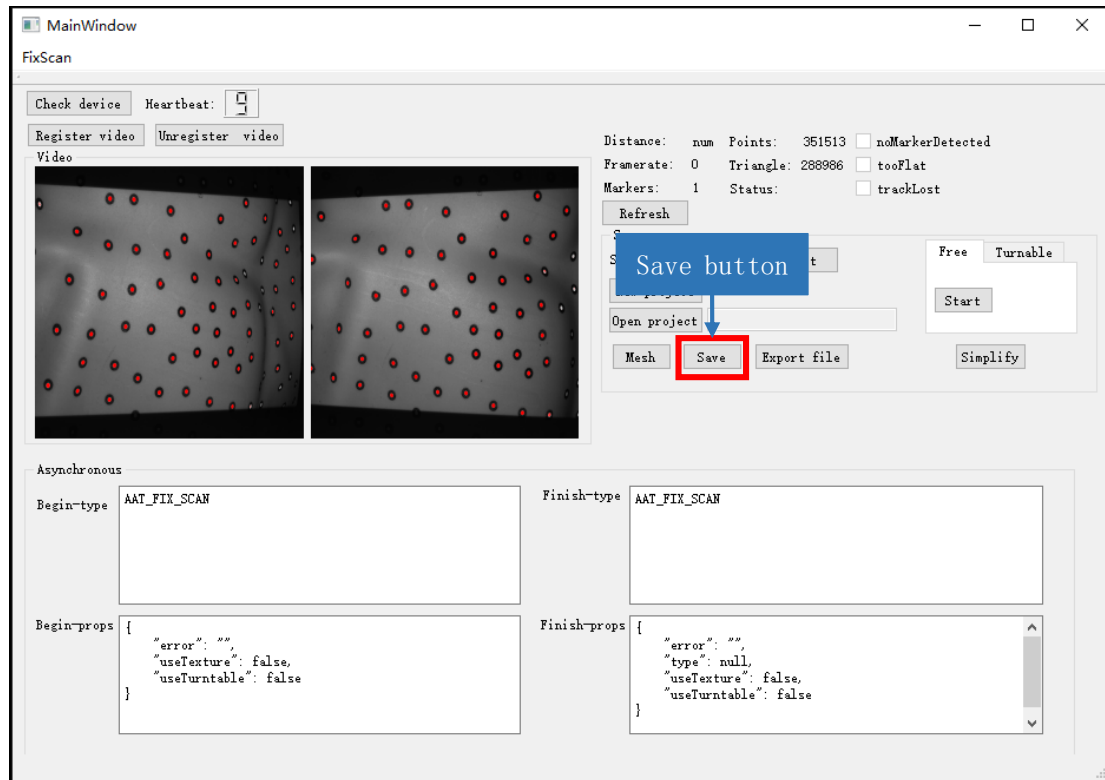


Figure 13 Save interface diagram

Table 6 Save processing parameters

Parameter	Implication	UI Type	Function
Path	Save route	Button+Text	save.cpp->on_pushButton_Path_clicked

P3、Asc、Sasc、 Std、Obj、Ply、 3mf	File save format	Check box	save.cpp->on_pushButton_clicked
OK	Confirm the save process	Button	save.cpp->on_pushButton_clicked

9、The meaning of the parameters corresponding to the red box in figure 14, the parameter details are shown in table 7.



Figure 14 The selected parameter of the box

Table 7 Parameter list

Parameter	Implication	UI Type	Function
Exit	Exit the scan	Button	mainwindow.cpp-> on_pushButton_ScanExitScan_clicked
Open project	Open project(.fix_prj)	Button+Text	mainwindow.cpp-> on_pushButton_ScanOpenProject _clicked
Mesh	Mesh package	Button	mainwindow.cpp-> on_pushButton_scanMesh_clicked
Save	Save project	Button	mainwindow.cpp-> on_pushButton_scanSave_clicked
Export file	Export file	Button	mainwindow.cpp-> on_pushButton_ScanExportFile_clicked
Simplify	Simplify	Button	mainwindow.cpp-> on_pushButton_scanSimplify_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.

10、 After the scan is completed, two project files will be generated in the selected path. The suffixes are “.fix\_prj”, “.rge” respectively. The corresponding model file will be generated in the path of the saved model.