Newland AIDC
Scanning Made Simple

Fixed Mount Barcode Scanner

NLS-Soldier180

User Guide

Disclaimer

Please read through the manual carefully before using the product and operate it according to the manual. It is advised that you should keep this manual for future reference.

Do not disassemble the device, doing so will void the product warranty provided by Fujian Newland Auto-ID Tech. Co., Ltd.

All pictures in this manual are for reference only and actual product may differ. Regarding to the product modification and update, Fujian Newland Auto-ID Tech. Co., Ltd. reserves the right to make changes to any software or hardware to improve reliability, function, or design at any time without notice. The information contained herein is subject to change without prior notice.

The products depicted in this manual may include software copyrighted by Fujian Newland Auto-ID Tech. Co., Ltd or third party. The user, corporation or individual, shall not duplicate, in whole or in part, distribute, modify, decompile, disassemble, decode, reverse engineer, rent, transfer or sublicense such software without prior written consent from the copyright holders.

This manual is copyrighted. No part of this publication may be reproduced, distributed or used in any form without written permission from Newland.

Fujian Newland Auto-ID Tech. Co., Ltd. reserves the right to make final interpretation of the statement above.

Newland Auto-ID Tech. Co., Ltd.
No.1, Rujiang West Rd., Mawei, Fuzhou, Fujian, China 350015
http://www.newlandaidc.com

Revision History

Version	Description	Date
V1.0.0	Initial release.	May 28th, 2025

Note: This document applies to firmware version NSet_V3.02.001.7 and above.

Contents

Chapter One: Basic Operation

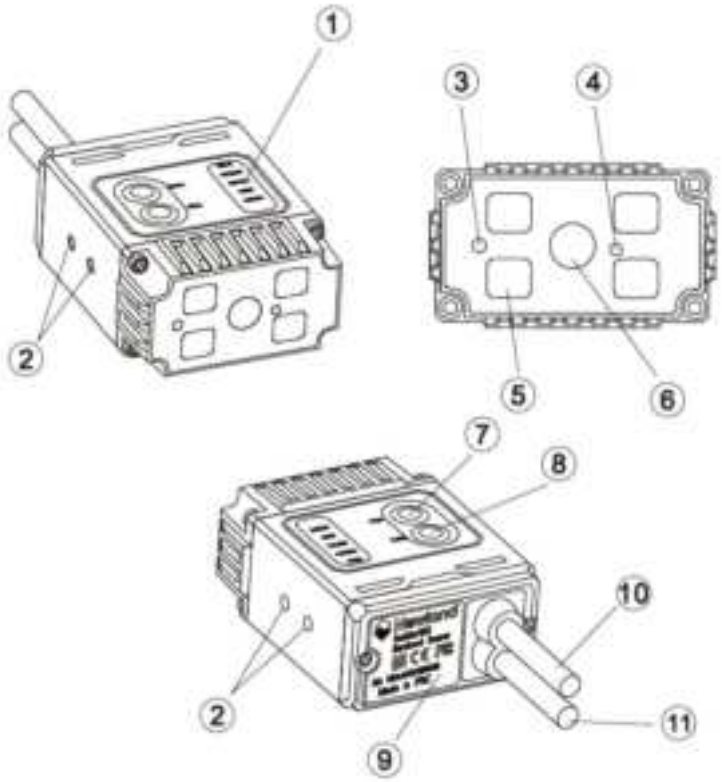
1 Preface

1.1 Unpacking

Model	Name	Package Contents	Quantity
NLS-Soldier180	Fixed Mount Barcode Scanner	NLS-Soldier180 Scanner	1
		Quick Start Manual	1
		Screws	4
		Mounting Bracket	1

1.2 Device Overview

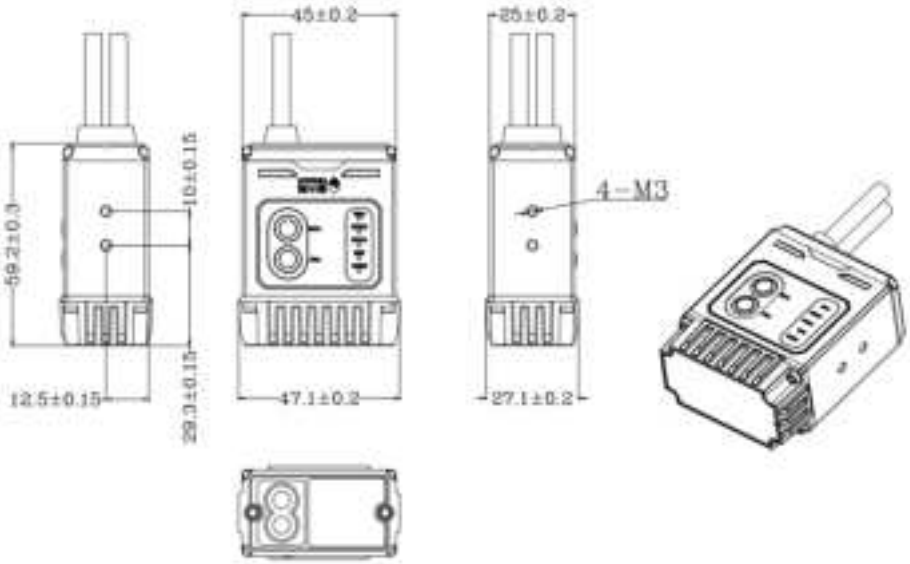
1.2.1 Product Appearance



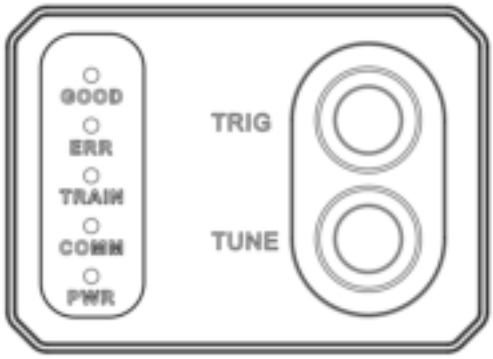
1	Function indicator	7	Trigger button
2	Mounting holes*4	8	Auto-learning button
3	Successful decoding LED	9	Label
4	Aimer	10	Data cable
5	Illuminating LED	11	Ethernet cable
6	Lens		

1.2.2 Dimension

Take standard device as an example: Units: mm

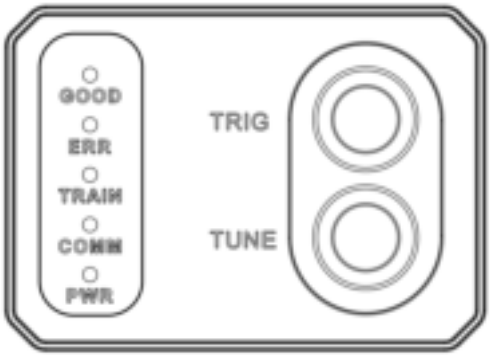


1.2.3 Button



Button	Setup	Operations
TRIG	Trigger	Short press once to start reading the barcode
	Barcode Programming	1. Long press for 9s until the buzzer beeps twice, then release to enter barcode programming mode. 2. Short press once again to exit. 3. Automatically exit after 60 seconds of inactivity. 4. Exits upon successfully reading other barcodes.
TUNE	Focus + Auto-learning	1. Long press for 3 seconds until the buzzer beeps, then release to enter Focus & Auto-learning mode. 2. Short press once again to exit Focus & Auto-learning mode.
TRIG+TUNE	Restore to factory settings	Long press for 20s to enter restore factory settings.

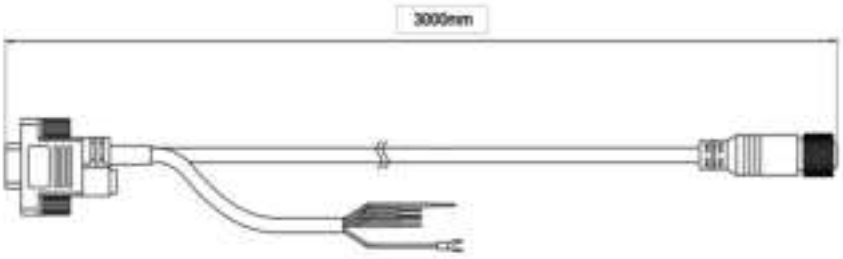
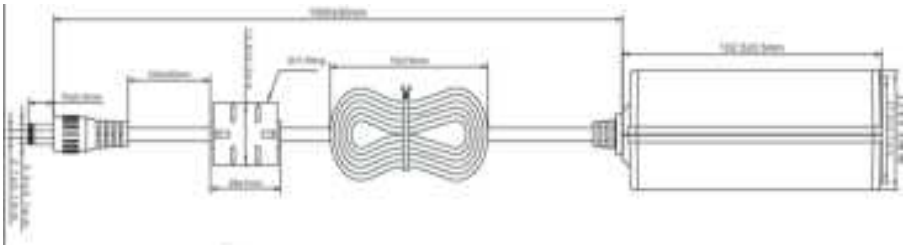
1.2.4 LED Indicator

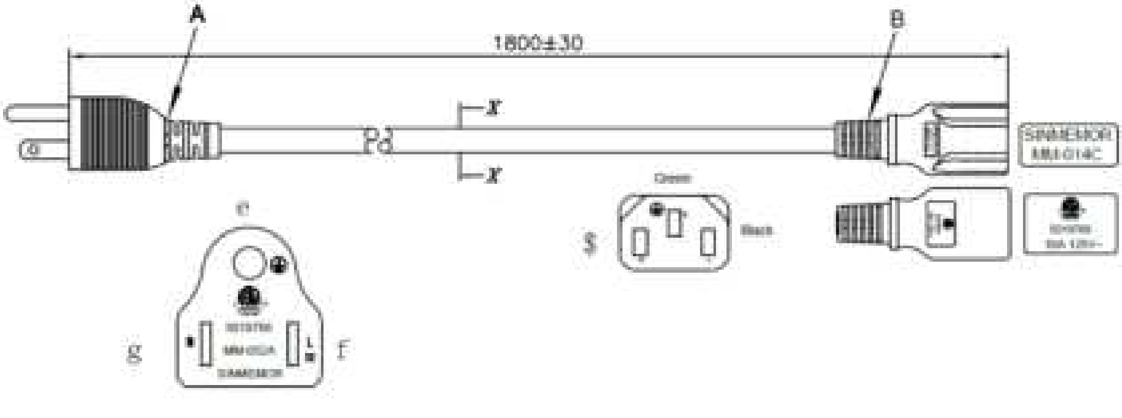


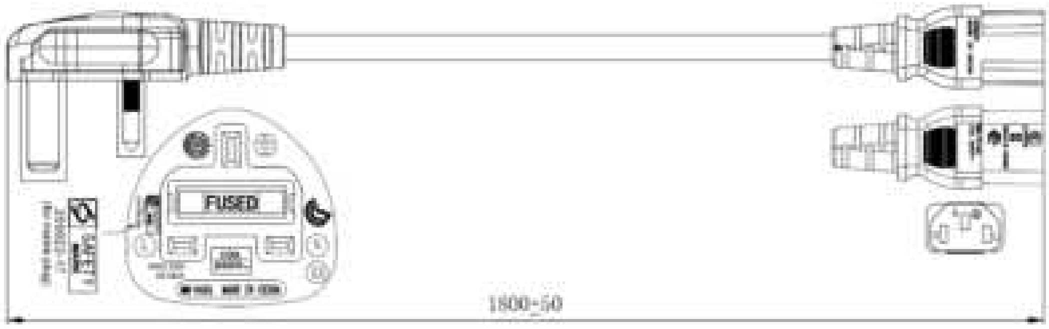
LED	Function	Description
GOOD	Good read indicator	Blinking Green: Decoding successful
ERR	Device error indicator	Blinking Red : Decoding failed Steady Red: An error has occurred on the device Possible causes include: 1.Communication error 2.CMOS abnormality 2. LED board malfunction 3.Firmware update failed 4. Abnormal communication with host during networking. 5.Script execution error
TRAIN	Auto-learning indicator	Blinking Orange : Auto-learning in progress
		Steady Orange : Auto-learning success
		Orange LED Off: Auto-learning failure
COMM	Communication indicator	Blinking Blue: Serial/Network data transmission
PWR	Power indicator	Steady Green : The device is powered on
		Green LED Off: The device is not powered on

2 Connection and Installation

2.1 Cable

Cable	Dimension (mm)
RS232+I/O cable	
Ethernet cable (5m)	
Adapter Cable	
Three-prong Power Plug	


A
B
1800±30
Pd
SINMEMOR
MM-014C
Green
Black
e
g
f

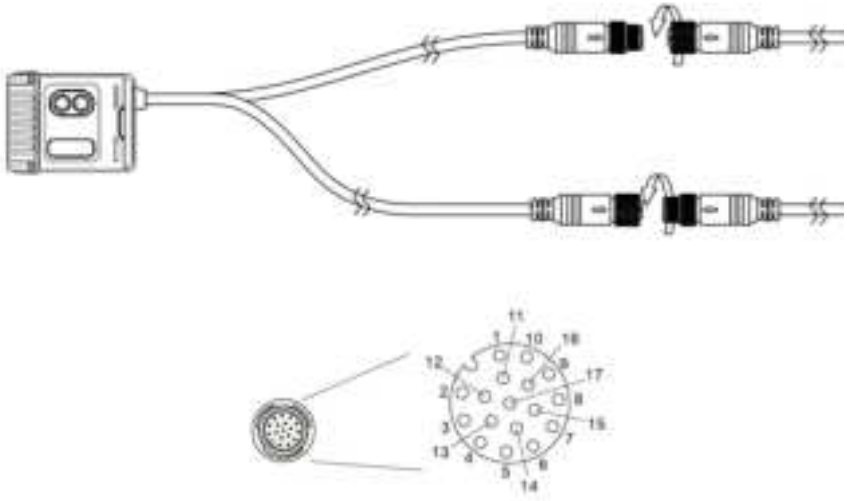

FUSED
1800±50


L
N
1800±50

Connect the Cable

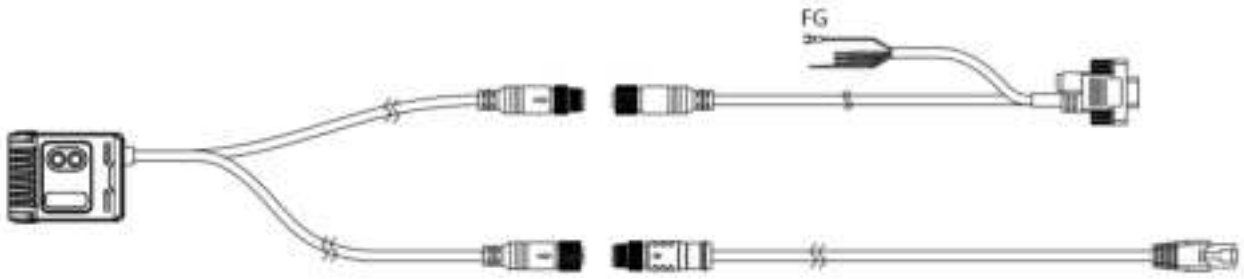
When connecting the cable, align the protruding part of the connector with the slot in the control port and insert it.

Turn the connector nut on the cable clockwise and tighten it.



Pin	Definition	Description
1	RS232-TX	Serial port sender
3	RS232-RX	Serial port receiver
4	GND	Power ground (black)
5	GND	Signal ground
6	I1A	Trigger signal input I1A(Brown)
7	I1B	Trigger signal input I1B(Pink)
8	I2A	Trigger signal input I2A(Green)
9	I2B	Trigger signal input I2B(Grey)
10	O1+	Signal output for good read O1+(Orange)
11	O1-	Signal output for good read O1-(White)
12	O2+	Signal output for not good read O2+(Yellow)
13	O2-	Signal output for not good read O2-(Purple)
17	VIN	VIN(Red)
Shell	FG	Frame ground (yellow-green)

Soldier180 Wiring

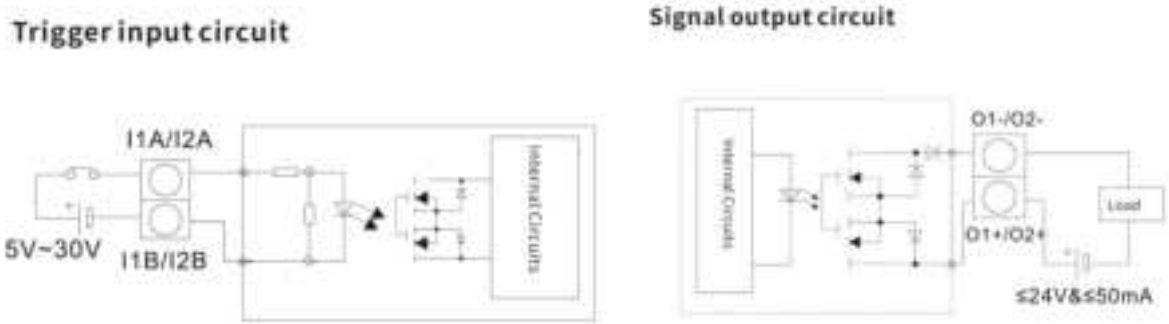


Note:

1. The FG wire should be grounded.
2. Always turn off the power before connecting or disconnecting the control cable.
3. The power wire (red) must not be exposed, as it may cause a short circuit.

External trigger input: when the target barcode passes through the scanning range of the scanner, the external equipment (such as sensor) sends a trigger signal to the external trigger input end of the scanner, and the scanner immediately decodes; in this way, fast automatic scanning and detection can be realized;

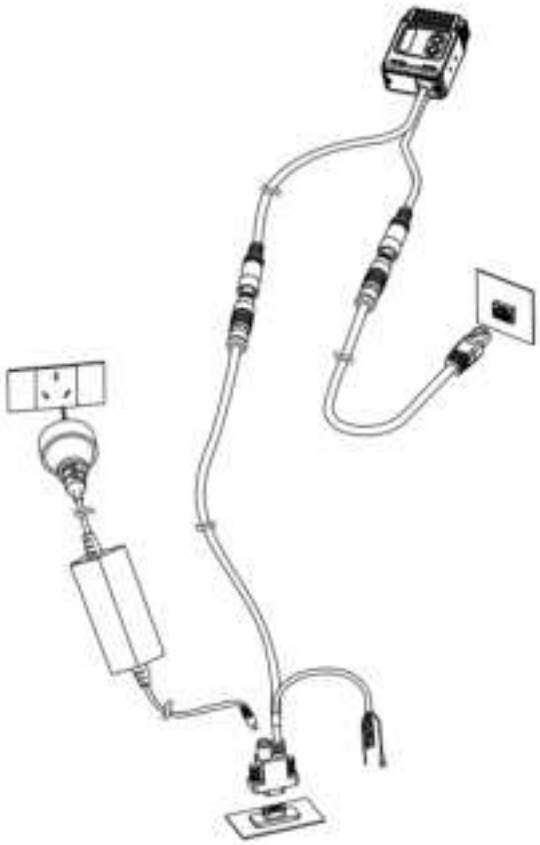
Signal output end: automatic detection also requires to be able to judge and indicate whether scanning is successful. Soldier180 provides two signal output ports for connecting with external indicating devices (such as light source, signal indicator, buzzer, counter), etc., O1 pin can be used to output the signal of scanning/decoding success, and O2 pin can be used to output the signal of scanning/decoding failure.



Note:

The electrical requirements of the pin for external trigger input are: 5V≤ input voltage ≤30 V. And the signal output, the supply voltage provided shall be ≤24 V. At the same time, a load resistor is required to ensure that the current does not exceed 50 mA.

2.2 Communicate with the Host Device



The scanner can be connected to the host device via RS232 or Ethernet.

Ethernet IP Settings:

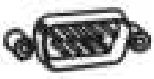
The default configuration of the Ethernet IP is:

IP: 192.168.1.100

Subnet mask: 255.255.255.0

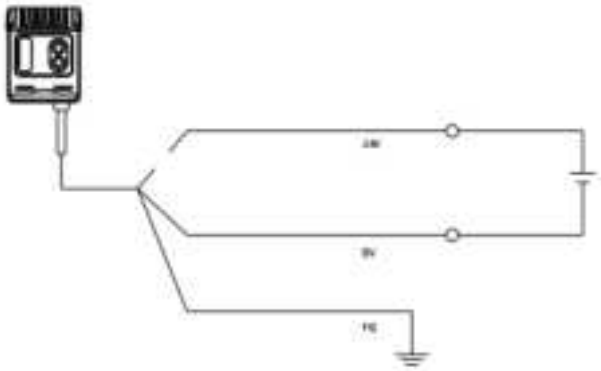
Gateway 192.168.1.1

Note: When the device is connected with the RS232+USB+I/O cable, the USB port does not provide power.

Interface	Location	Image
RS232	RS232 port on the host device	
RJ45	RJ45 port on the host device	

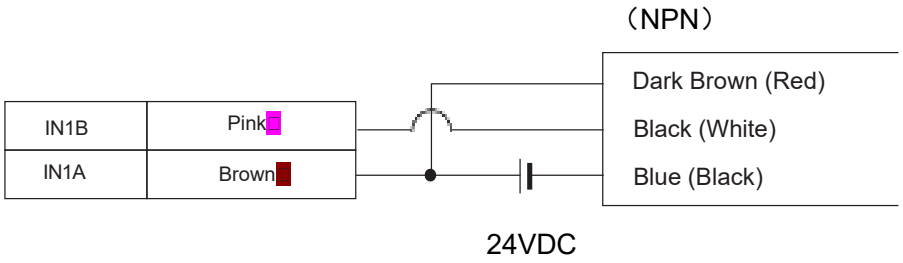
2.3 Sensor and External Switch Connections

Direct Power Wiring (without Adapter)

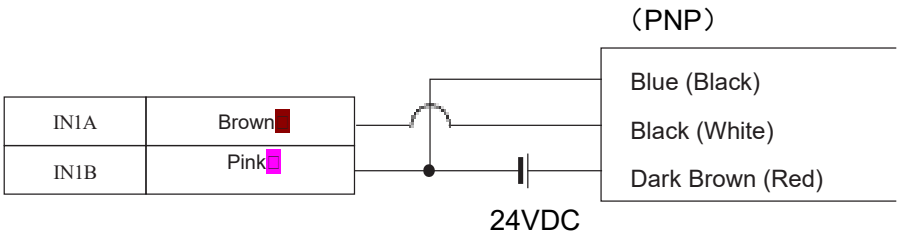


Photoelectric Sensor Wiring

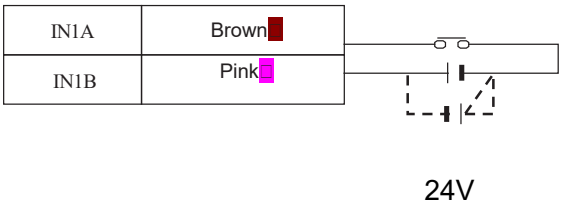
NPN



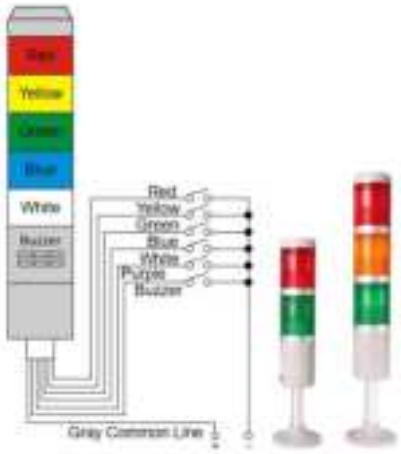
PNP



Switch Wiring



Alarm Indicator Wiring



Alarm Indicator Wires	Soldier180 Wires
Green	O1+ (Orange)
Red/Purple	O2+ (Yellow)
Gray (Common)	VIN (Red)
-	GND (Black), O1- (White), O2- (Purple) – All connected together

Note:

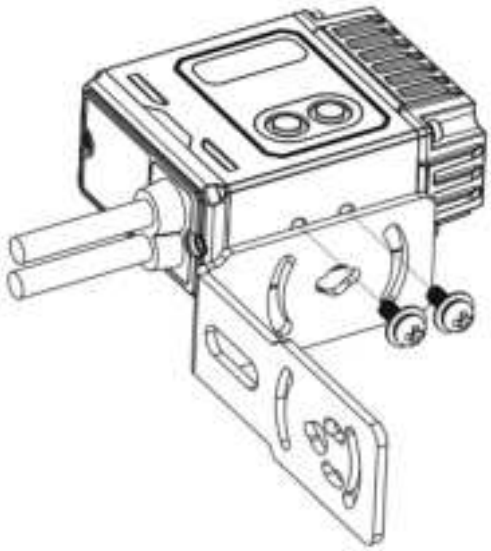
When the barcode is successfully read, the green LED turns on.

When the barcode reading fails, the red LED turns on and the buzzer beeps.

2.4 Installation

Installation without Bracket

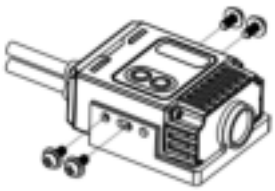
4 screw holes are reserved on the back of the device, choose the appropriate location to install the device according to your needs



Installation with Optical Accessories

Optional accessories such as a diffuser, magnifying lens, and reflector can be installed on the Soldier180 depending on the application.

Diffuser Installation



Magnifier Installation

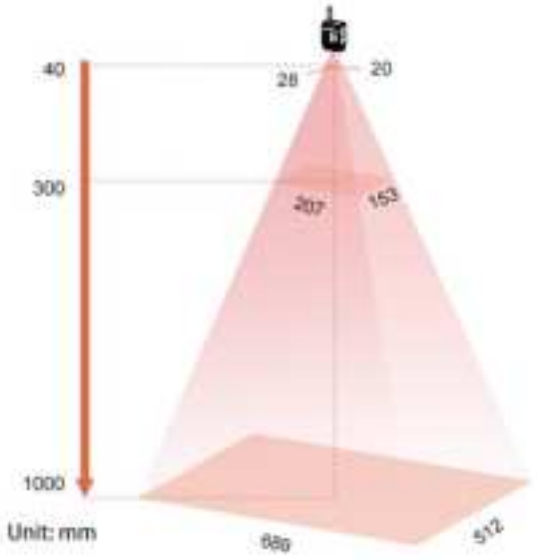
Reflector Installation

2.5 Installation Position Adjustment

- **NLS-Soldier180N-P**

Unit: mm

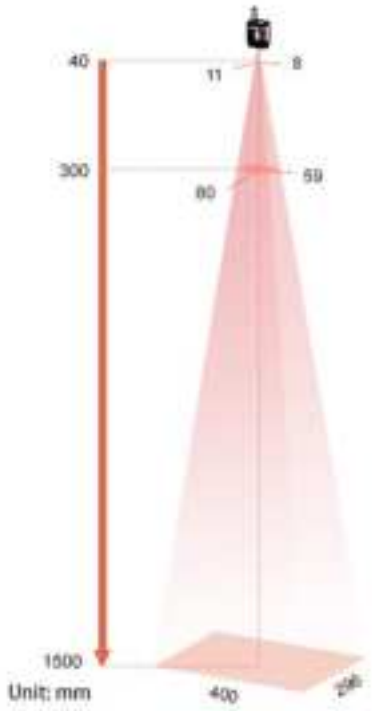
Barcode Type	Barcode Size	NLS-Soldier180N-P(EFL=6.8) Installation Height	The Horizonal Field of View	The vertical Field of View
1D	2mil	65	44	33
	4mil	150	103	77
	11mil	500	344	258
2D	2mil	40	27	20
	5mil	105	72	54
	10mil	225	154	116
	16mil	375	258	193
	43mil	1000	688	516



- **NLS-Soldier180S-P**

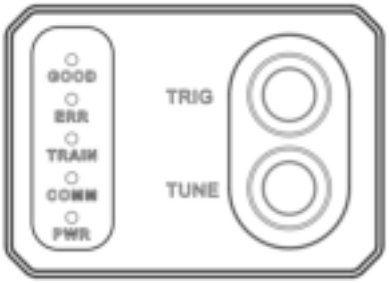
Unit: mm

Barcode Type	Barcode Size	NLS-Soldier180S-P(EFL=13.7) Installation Height	The Horizonal Field of View	The vertical Field of View
1D	1mil	65	17	13
	4mil	375	100	75
	9mil	1000	266	200
2D	0.7mil	40	10	8
	3mil	150	40	30
	4mil	225	60	45
	9mil	500	133	100
	25mil	1500	400	300



2.6 Auto-learning

The NLS-Soldier180 supports auto-learning function. Long-pressing the TUNE button on the device to enter auto-learning mode.



Button	Function	Operations	Status Indicators
TUNE	Focus+Auto-learning	1.Long press for 3 seconds until the buzzer beeps, then release to enter Focus & Auto-learning mode. 2.Short press once again to exit Focus & Auto-learning mode.	In Focus/Auto-learning mode The TRAIN LED is blinking orange.
			Focus/Auto-learning is completed The buzzer beeps once, and the TRAIN LED stays orange.
			Focus/Auto-learning is failed TRAIN LED is not on

Button-triggered self-learning results are automatically saved to the designated Configuration Library (Library 8). Each new learning cycle will overwrite the prior configuration data in this library.

Chapter Two: Nset

3 NSet Introduction

3.1 NSet Introduction

NSet is a User Interface client application developed by Fujian Newland Auto-ID Tech. Co., Ltd.

That simplifies initial scanner setup and changing parameters of the scanner you use.

Its main functions include:

- Debug equipment parameters and take effect in real time, view scanned images and barcode reading data in real time ;
- Upgrade the firmware, import the configurations, restore to the factory default ,etc in batch;
- Provide configuration comparison between multiple devices to analyze the performance differences of different devices;
- Generate the setting barcodes for the configuration of the scanner, and save them as a PDF or Word file;
- Interact with the device through commands;
- Provide quick selection tools to help users quickly assess scene requirements, precisely select the model and determine the installation location

3.2 Installation Guide

NSet software supports Windows XP/7/8/10 operating system.

There is no need to install the driver additionally, just double-click the NSet software installation file and follow the step-by-step instructions to complete the installation.

Recommend installing NSet_V3.02.001.7 and above software.

3.3 Nset Home

The NSet Home screen consists of the following main sections:

1. Main Menu
2. Device List
3. Device Information

Note: Click the "Download Center" in the upper-right corner to visit the official website and download the latest documents.

3.4 Network Setup

The default configuration of Ethernet IP is as follows:

IP address: 192.168.1.100

Subnet mask: 255.255.255.0

Gateway address: 192.168.1.1

After selecting a device, click the "**Network Setup**" button to bring up an interface that displays the IP address of the local machine (PC). To ensure the scanner is on the same subnet as the PC, you can either enable **DHCP** or configure the settings manually.

Method 1: In the device list, select the device you want to configure, right-click the settings icon, and select the "Network Setup".

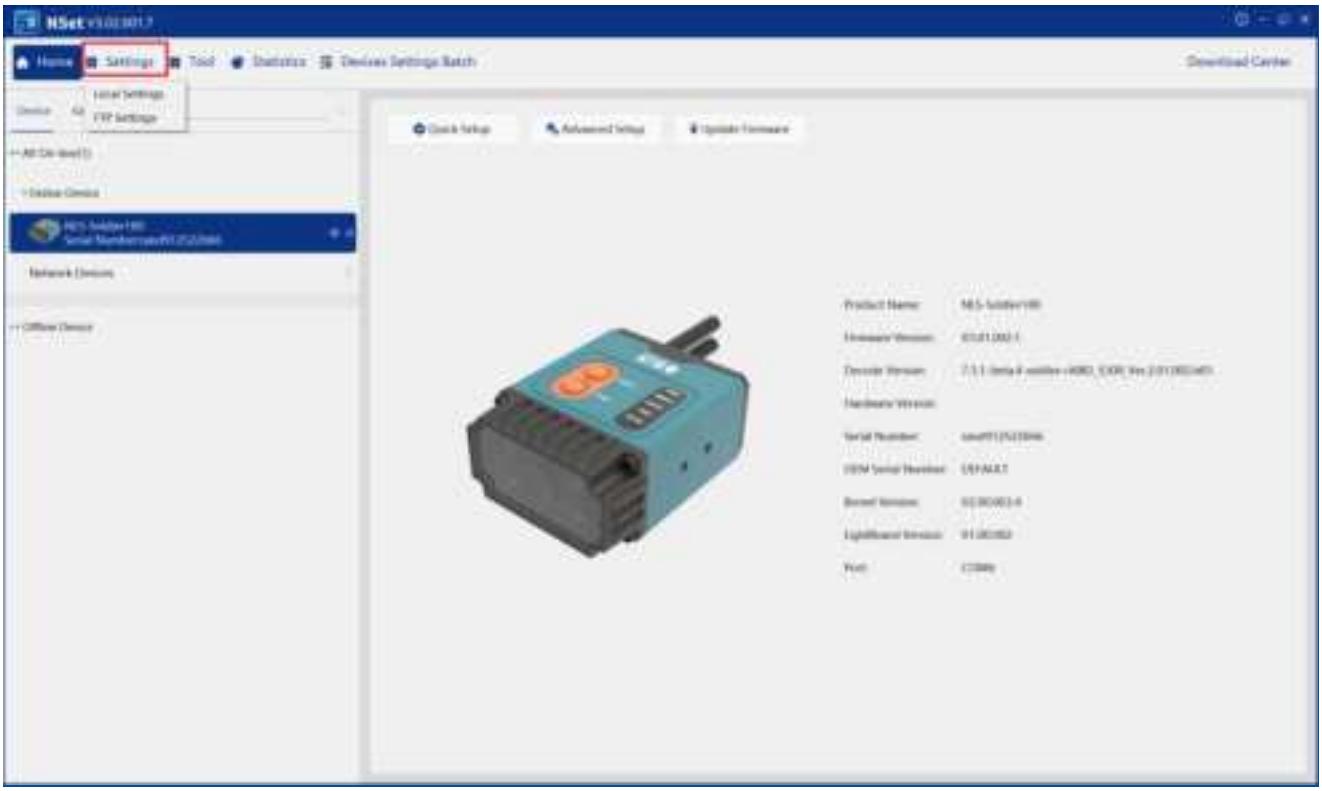
Method 2: In the device information panel on the right, click "Network Setup".

2. In the Network Setup window, you can either enable DHCP to obtain an IP address automatically, or manually configure a fixed IP address based on the host IP.

After setting the IP, double-click the device to enter the main operation interface of the device.

3.5 Main Menu

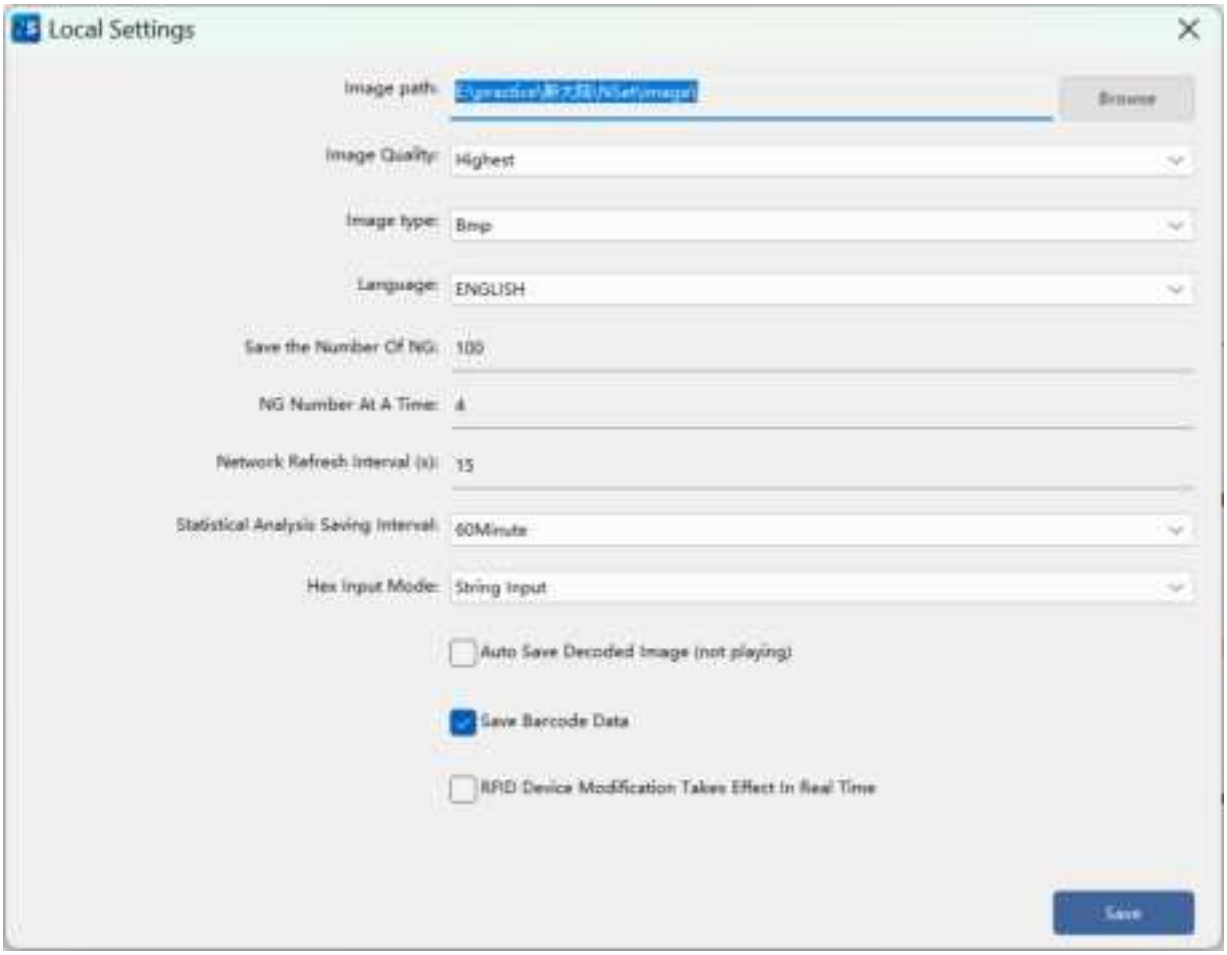
3.5.1 Settings



Local Settings

The Local Settings menu shows parameters that allow you to select which image quality/type and set where images captured by the scanner are stored. Here, you can also select the language (Chinese or English) of Setup Tool itself.

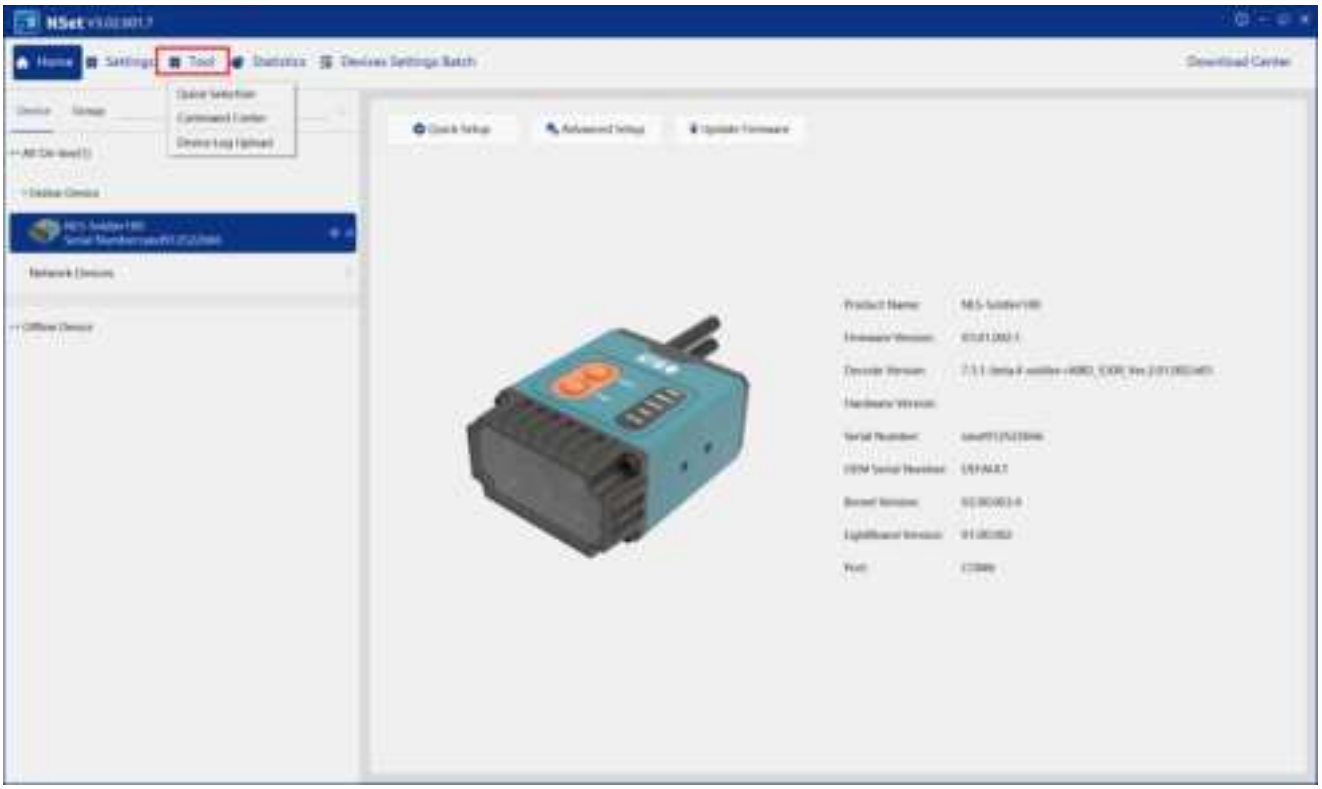
"Save the number of NG" is the upper limit of the image failed to decode.



FTP Settings

Below interface is used to configure the FTP server on the PC, with the device functioning as an FTP client.

3.5.2 Tool



Quick Selection

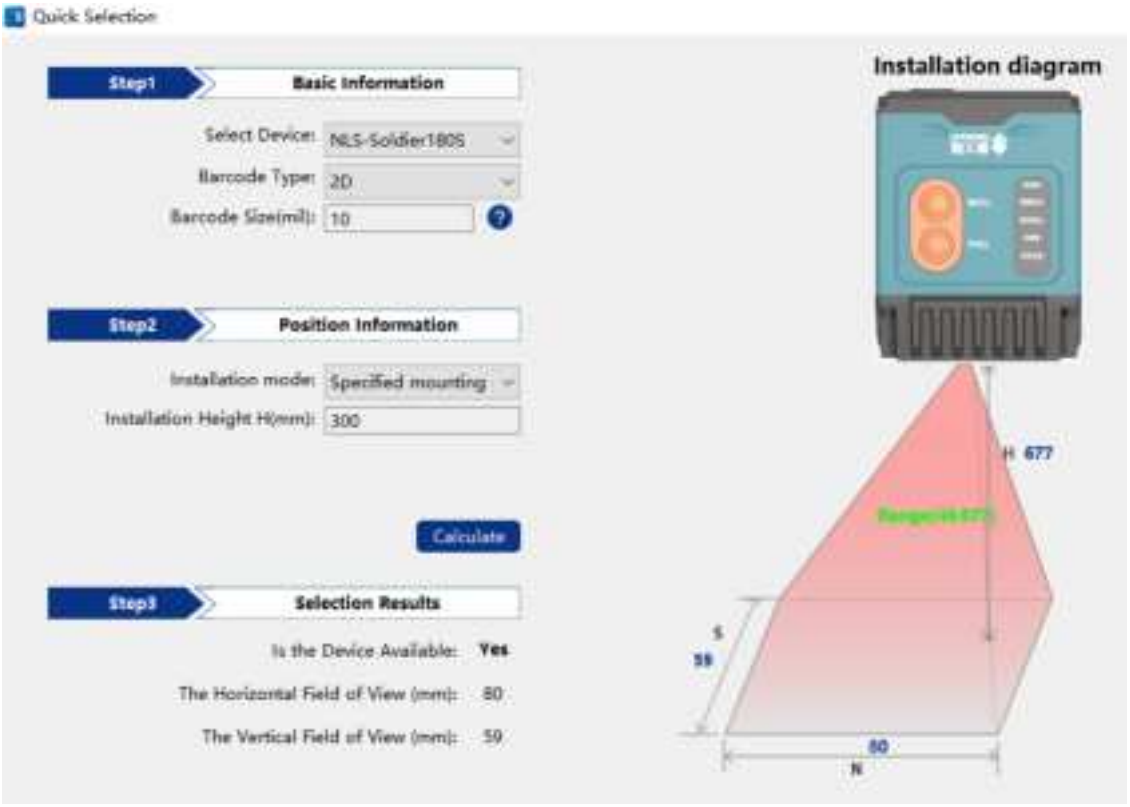
This function can be used to evaluate whether a device meets the application requirements by analyzing barcode parameters, installation constraints, and field of view (FOV) specifications

Step 1: Select the device model, barcode type, and barcode size.

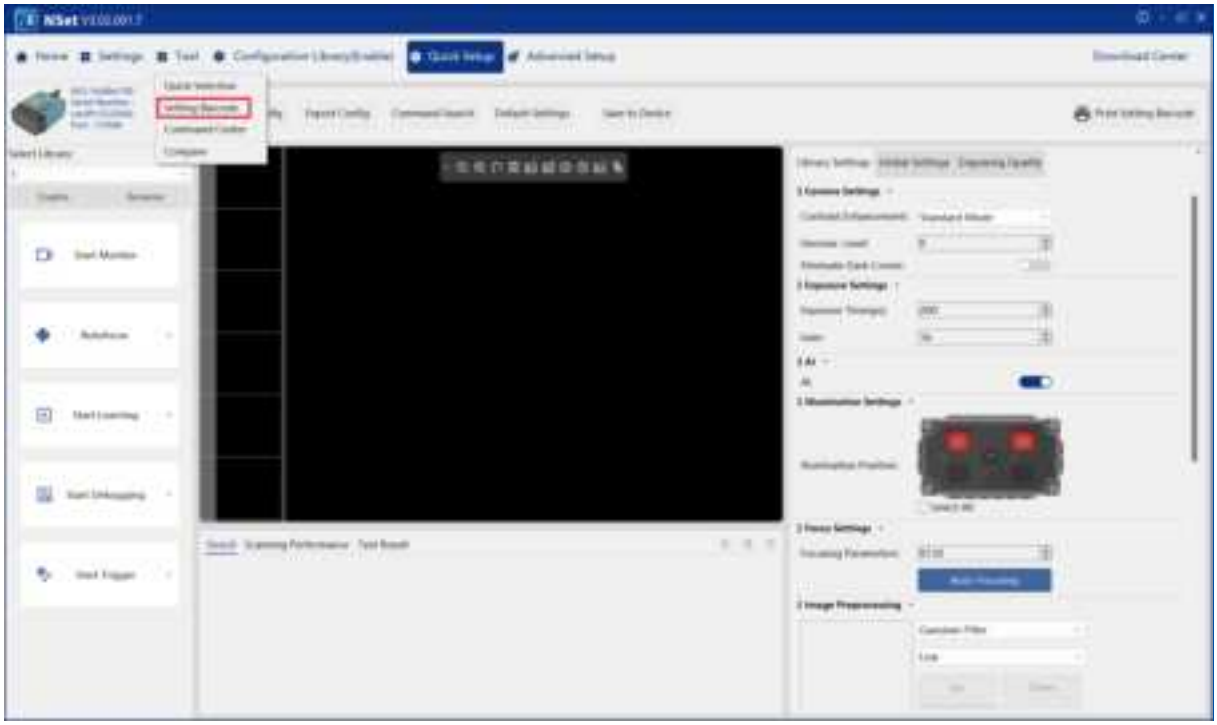
Click the  button to open the mil conversion tool, which automatically converts the entered barcode data into mils.

Step 2: select the installation mode and enter installation height.

Click "Calculate" to determine whether the selected device meets the application requirements and view the installation diagram.



Setting Barcode



Generate Setting Barcode

There are two ways to generate setting barcodes:

1. Generate the setting code based on the configuration of the modifications compared to the last save.

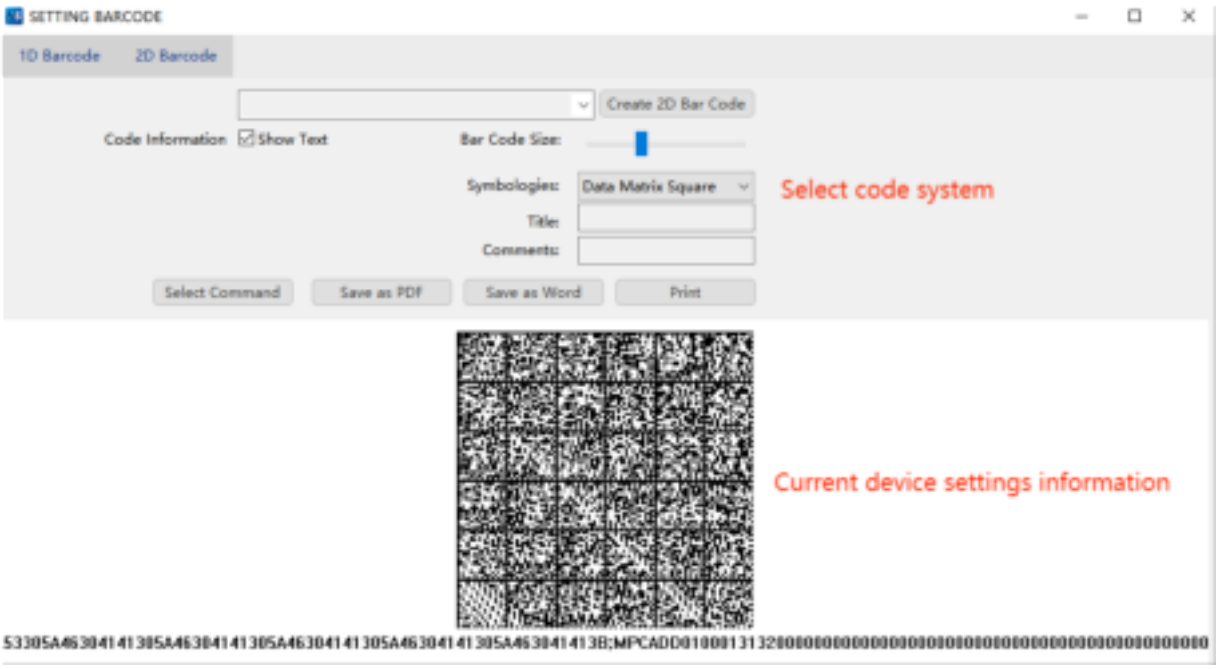
2. Generate setting barcodes by inputting instructions.

Generate setting barcode by inputting instructions

Generate setting barcodes based on modified configuration

1. Modify the configuration parameters as needed.

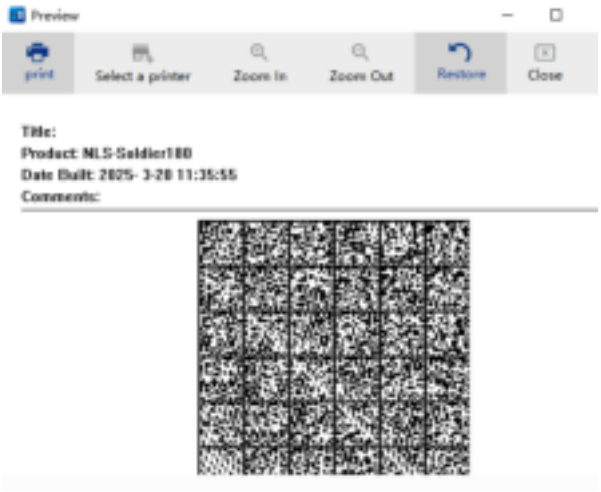
2. Click the "Setting Barcode" button in the toolbar to generate setting barcode based on the updated configuration.



Print Setting Barcode

After generating the setting barcode, click the 

to print the setting barcode.



Read the Setting Barcode

To read setting barcodes, press the TRIG button to enter barcode programming mode. The device will automatically perform illumination and focus adjustments. Before scanning, adjust the device's position using the focus indicator to ensure proper alignment.

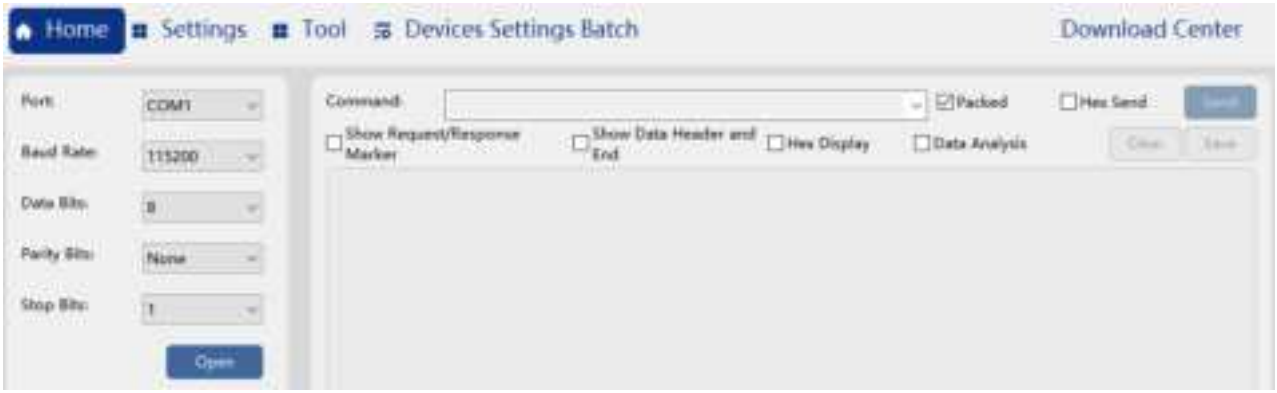
Setup	Operations	Device Status	Status Indicators
Barcode Programming	1. Long press for 9 seconds until the buzzer beeps twice, then release to enter barcode programming mode. 2. Short press once again to exit. 3. Automatically exit after 60 seconds of inactivity. 4. Exits upon successfully reading other barcodes.	In barcode programming mode	The Good Read LED is continuously blinking green
		Successfully read the setting barcode	Buzzer beeps once

Command Center

On the main interface, select the port where the device is located (the device needs to be set to serial port, USB CDC or HID POS), after opening the serial port, you can interact with the device through commands. If you enter the Advanced Setup interface of the device, click "Options" - "Command Center", and enter the corresponding instruction commands.

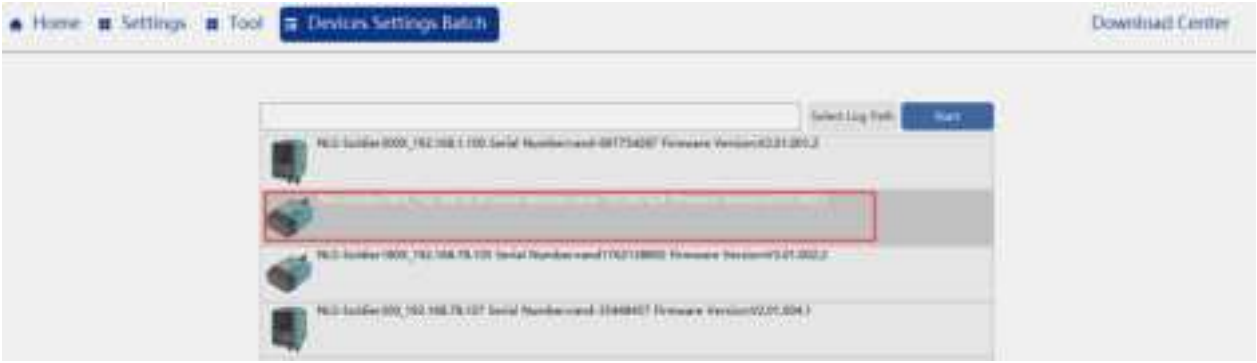
If "Packed" is selected, the packet header and tail will be added according to the instruction structure and sent; if it is removed, the original data will be sent.

Select "Data Analysis" will parse the barcode uploaded by the device to the PC according to the packaging format selected by the user (command is: "PACKAG", if the device supports it).



Upload Device Log

Click "Select Log Path" to choose the log directory, then click "Start" and select the target device to export the logs. The device logs are used for troubleshooting and failure analysis.

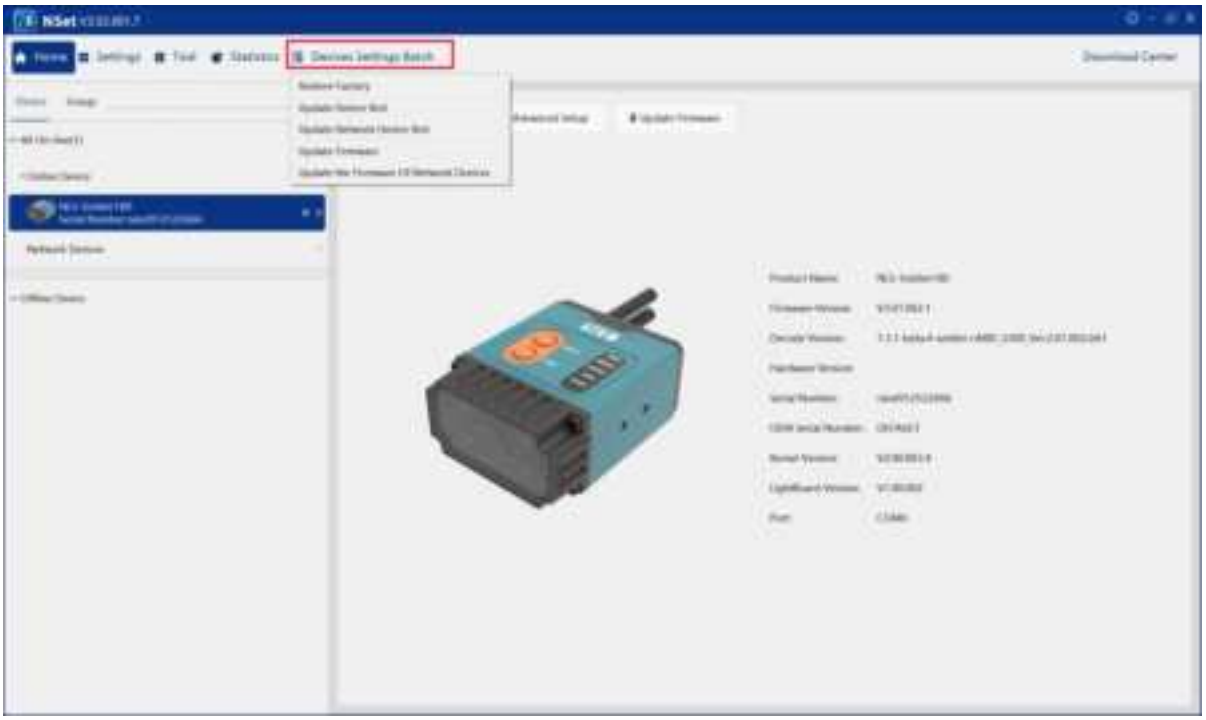


3.5.3 Statistics

Select devices to be monitored. The system supports monitoring multiple devices simultaneously.

3.5.4 Device Setting Batch

In the devices setting batch section, the following functions are available: "Restore Factory", "Update Device Xml", "Update Network Device Xml", "Update Firmware", and "Update the Firmware of Network Devices".
The "Restore Factory", "Update Device Xml", and "Update Firmware" are used for devices with USB interfaces.
For devices that communicate via a network, use “Update Network Device Xml" and "Update the Firmware of Network Devices" to update the configuration file and firmware.



Restore Factory Defaults

Enable "User confirms to restore factory default". A confirmation message will pop up each time you reset the device

Select "Operate current device" to restore the currently connected device to the factory defaults.

Click "Start", the NSet will automatically identify the device connected and start to reset the device.

It can support the simultaneous operation of multiple devices.

Update Device XML

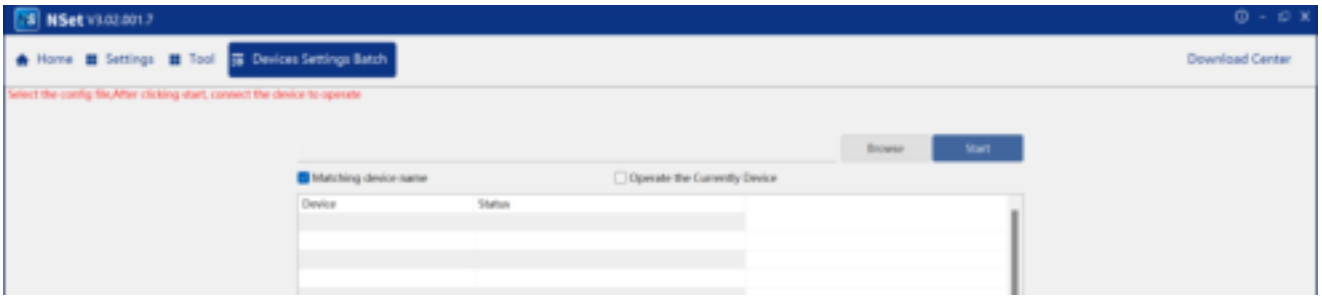
Click "Browse" to select the configuration file to update.

Select "Matching Device name", it will be matched according to the device name, and the device with the same name will be updated. If it is not matched, it will fail to update the configuration file.

Select "Operate the Currently Device" to update the configuration file of the currently connected device.

Click "Start", the NSet will automatically identify the device access and start updating the configuration.

It can support the simultaneous operation of multiple devices.



Update Network Device XML

This function allows batch updating of configuration files for devices on the local area network (LAN).

Update Firmware

Click "Browse" to select the firmware file to be updated.

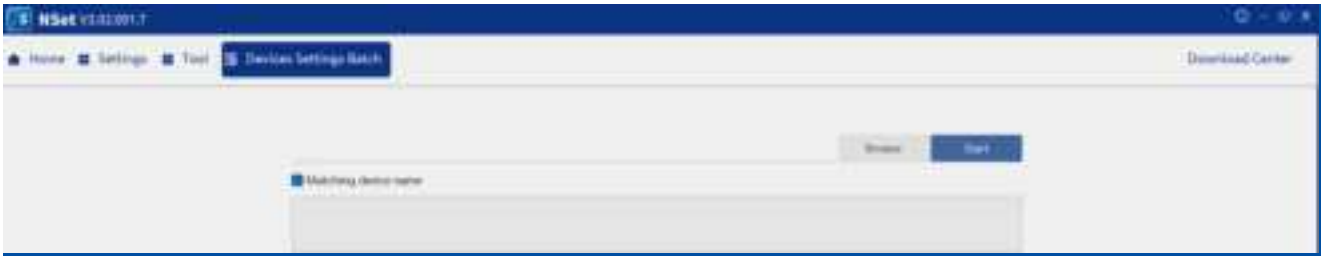
Select "Matching Device name", it will be matched according to the device name, and the device with the same name will be updated. If it is not matched, it will fail to update the configuration file.

Click "Start", select the device or re-connect the device to start the update.

It can support the simultaneous operation of multiple devices.

Update the Firmware of Network Devices

This function allows batch updating of firmware files for devices on the local area network (LAN).

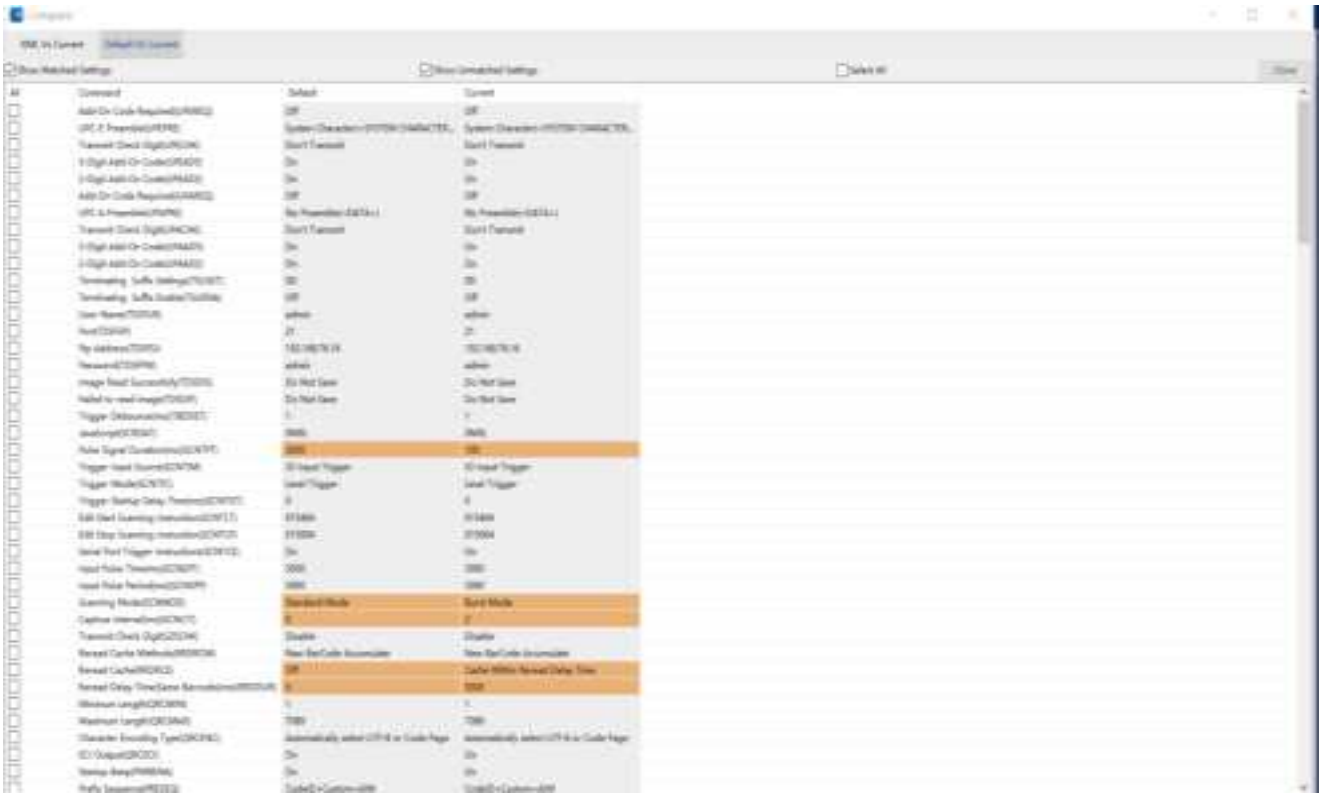


Configuration Compare

Select the device to be compared in the device tree on the left, and click "Start" to list the configuration differences between the devices.

Select "Show Matching Settings" will list the same configuration between devices.

Select "Show Unmatched Settings" will list the different configurations between devices, and marked in orange.

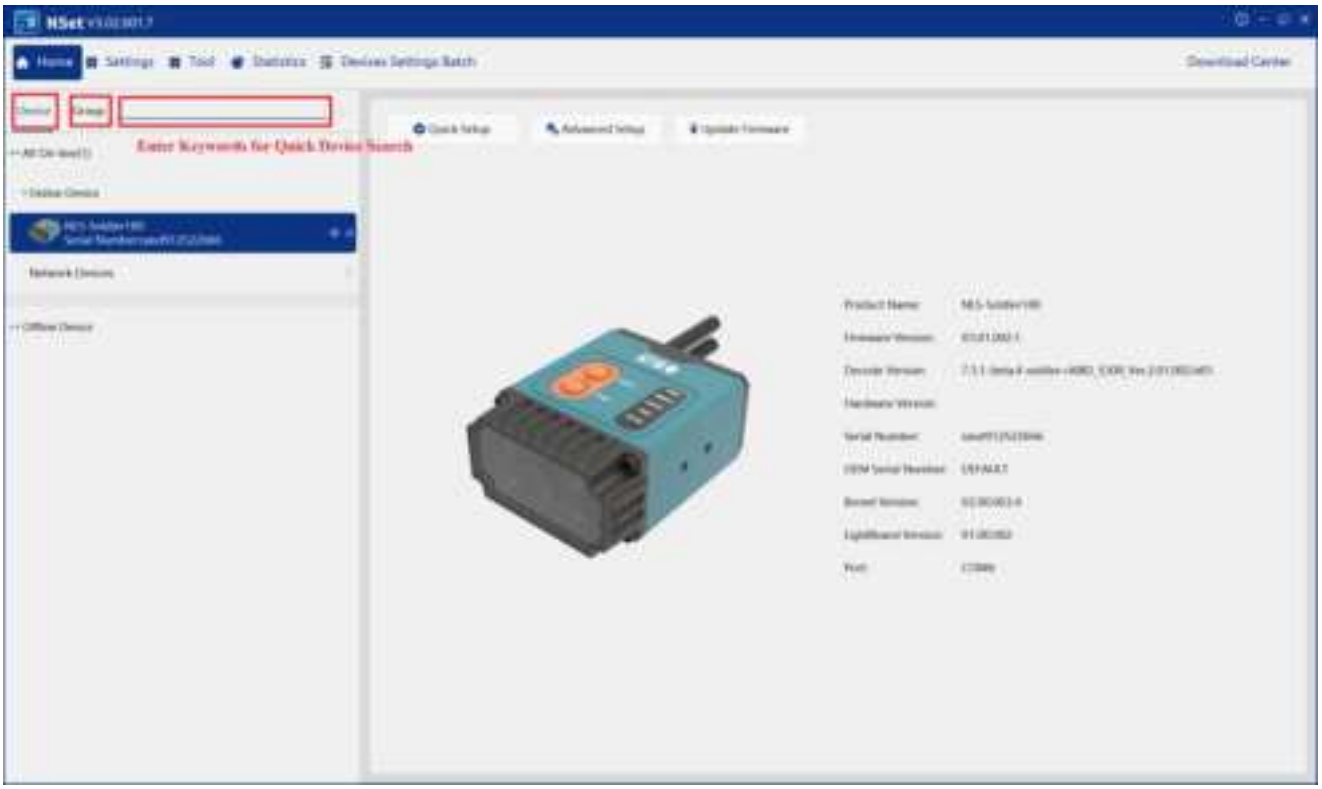


3.6 Device List

The device list on the left is categorized into: "Device", "Group" and "Device Search Box".

The "Device" section includes: "All On-line" and "Offline Devices”, where you can view the devices that are currently online and offline.

In the "Device Search Box," you can enter keywords to quickly search for devices.



3.6.1 Device Connection

The "Device" section includes: "All On-line" and "Offline Devices”.

All On-line

NSet will automatically display all connected devices, and when a device is selected, the system information of the device will be displayed on the right side of the home page, including: product name, hardware version, algorithm library version, serial number, date of manufacture, IP address and mac address and other information.

Among them, the devices connected by Ethernet are unified into a list of "Network Devices", and the rest are listed as "Online Device".

If you find that any device is not displayed in the "All On-line" Column, please click "  " At the top right to refresh and try again;

Or click "  " On the right side of the "Network Devices" Column to manually add a network device.

Offline Device

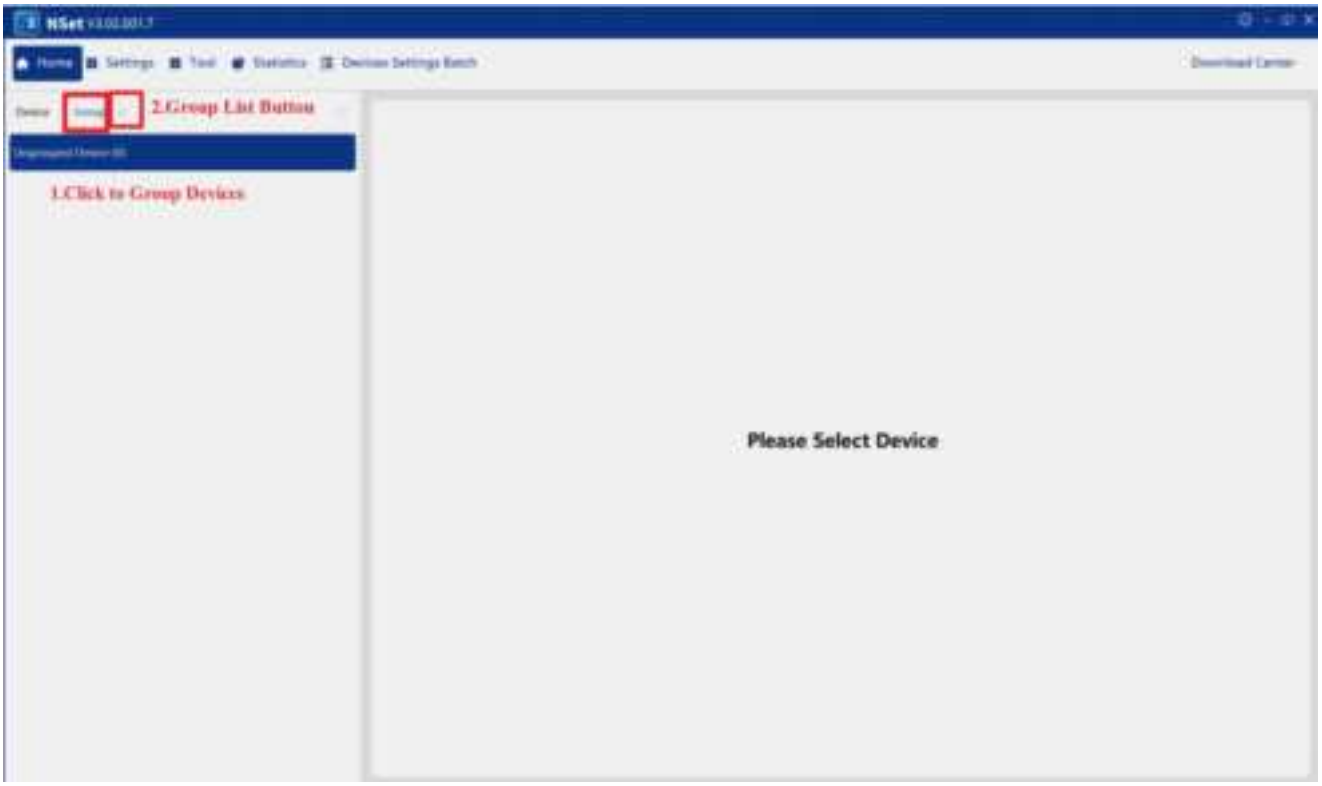
When the device is not connected, you can view the offline parameters of industrial products, including name and offline configuration version number.

3.6.2 Device Group

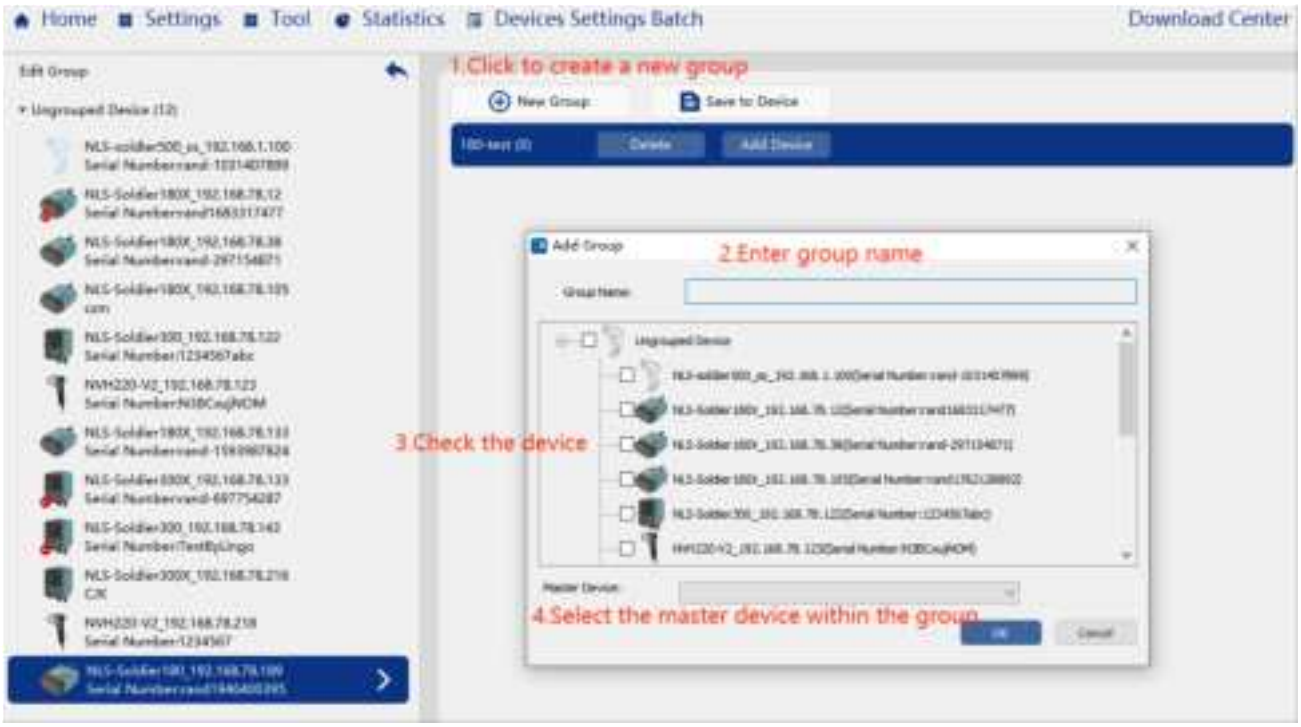
Devices connected through Ethernet can create device groups to meet the multi-device combination barcode scanning in complex scenarios such as multiple fields of view, large fields of view, and multiple depths of field.

"Device Group" Is only applicable to NVF and Soldier series products configured with Ethernet interface.

Click "Group" at the top of the device list, switch to the device group list, and click "" on the right; or click "Edit Group" in the "General setup"; enter the device group editing interface.



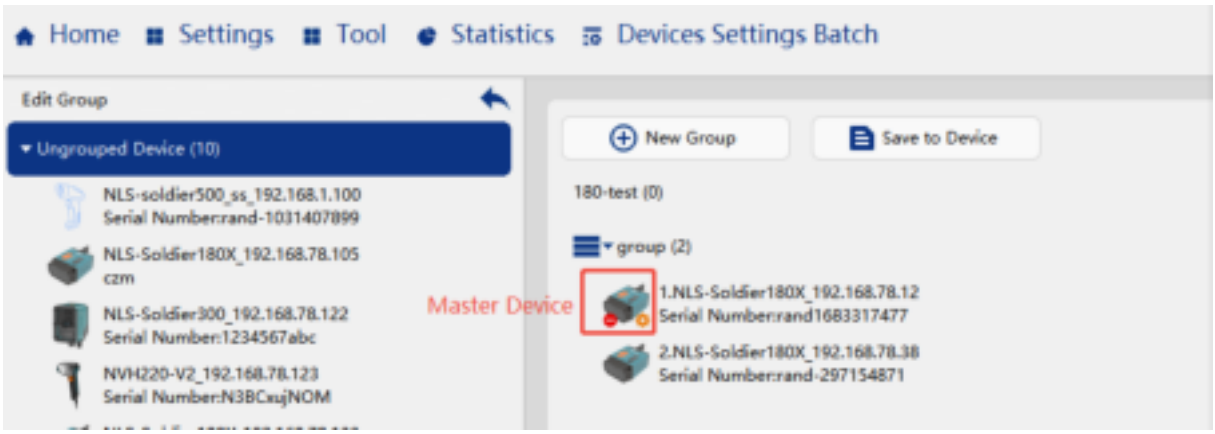
Click "New group", the group setup are as follows:



Name	Function
Group Name	Customize the group name
Networking Device	Select and add devices that need to be grouped
Master Device	Select a master device for the group

Click the specific device in each group, and the setup items on the right include: "Master Device", Delete, "Move down" and "Move up", among which the main device also has the "Edit (group) Properties".

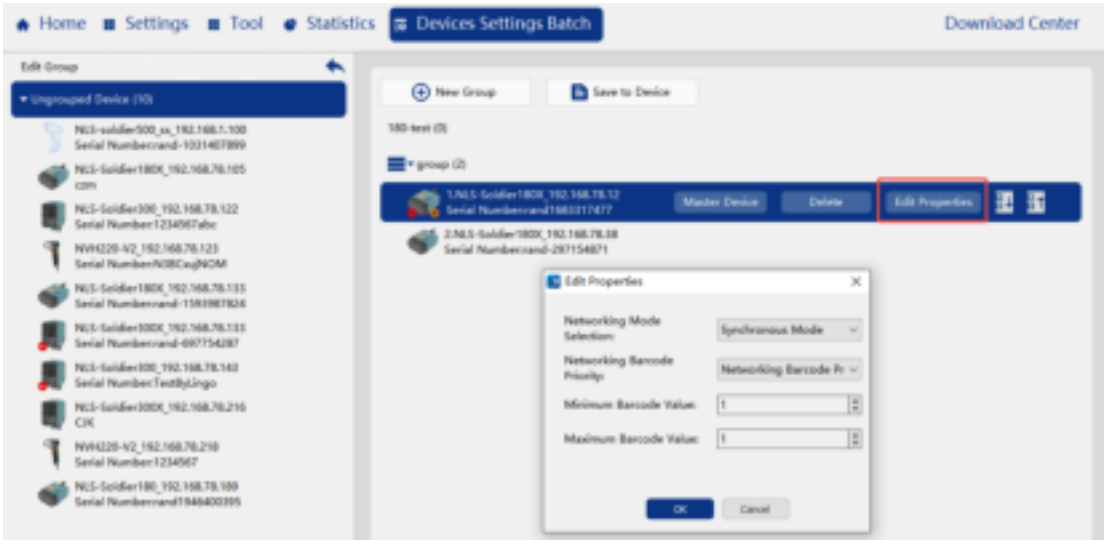
The icon of the main device is marked with a "⚙" in the lower right corner of the icon.



Click a specific device in the "Ungrouped Device" list, and click ">" on the right to add the device to the specified group.

Click "Save to Device" to save the group setup.

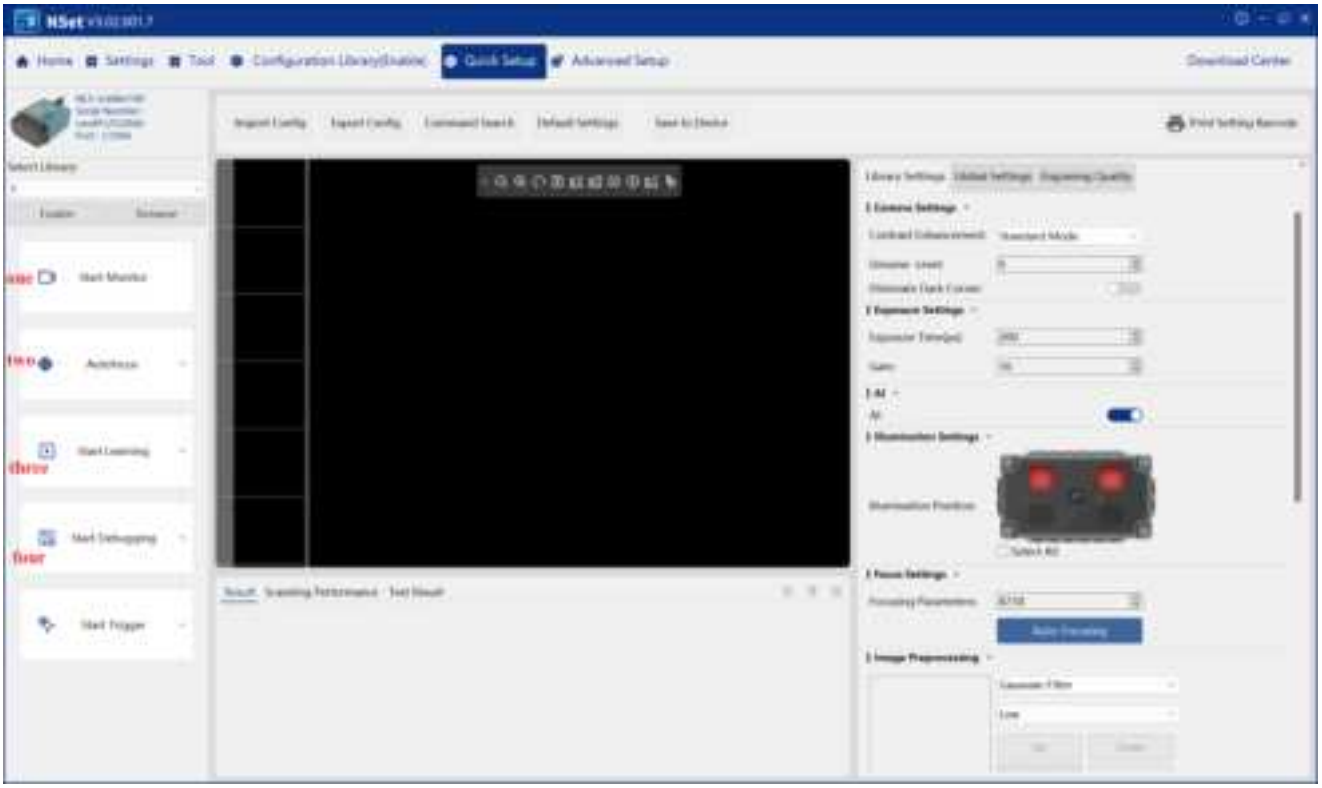
After the new group is completed, the group attribute can also be modified on the main device.



Name	Function
Networking Mode Selection	Synchronous Mode only
Networking Barcode Priority	Networking Barcode Priority Mode only
Minimum Barcode Value	The minimum and maximum number of the networking barcodes. The two values can be the same.
Maximum Barcode Value	

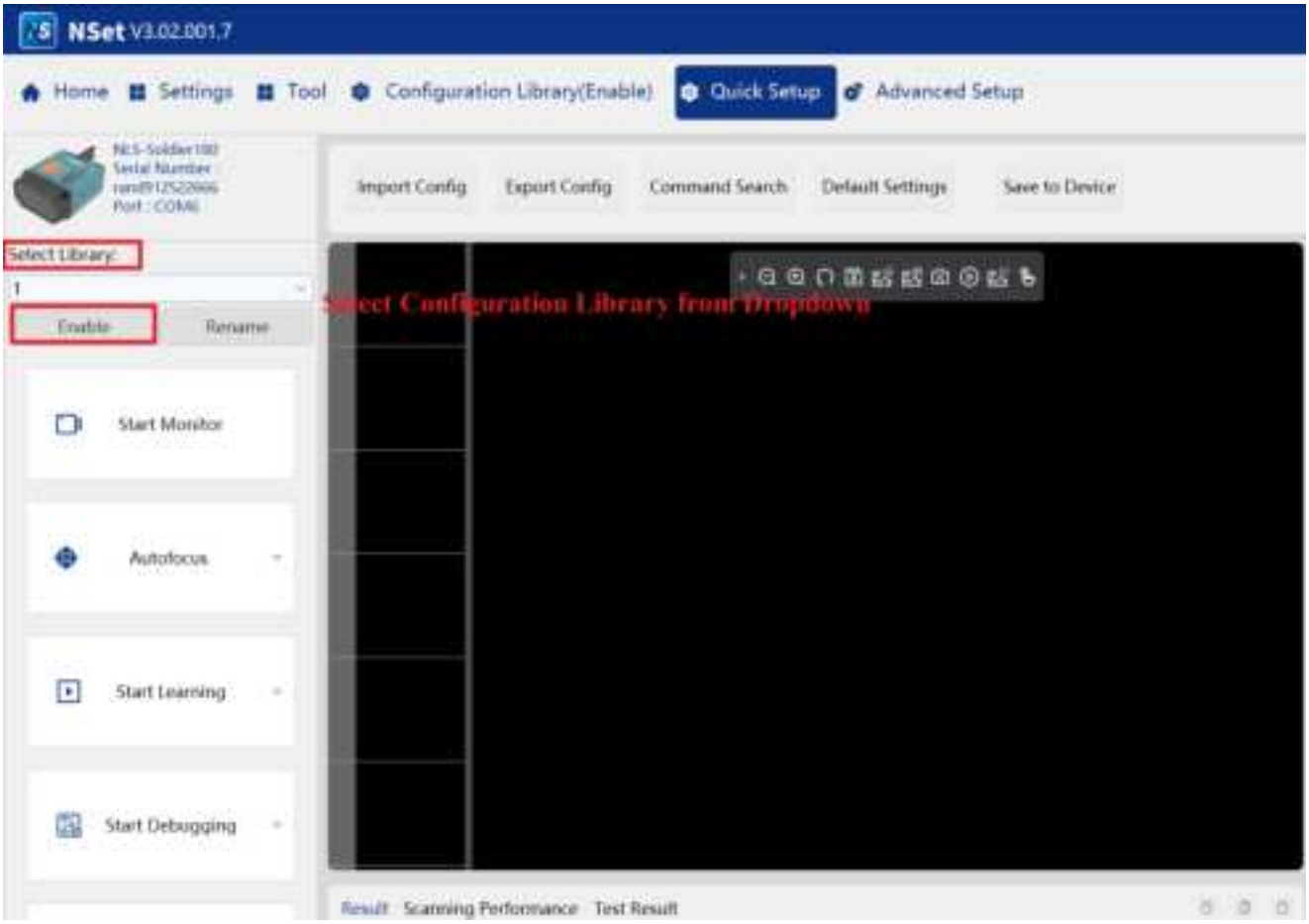
4 Quick Settings

"Quick Setup" is based on auto-focus and self-learning functions, help users to quickly complete the automatic configuration of device parameters.



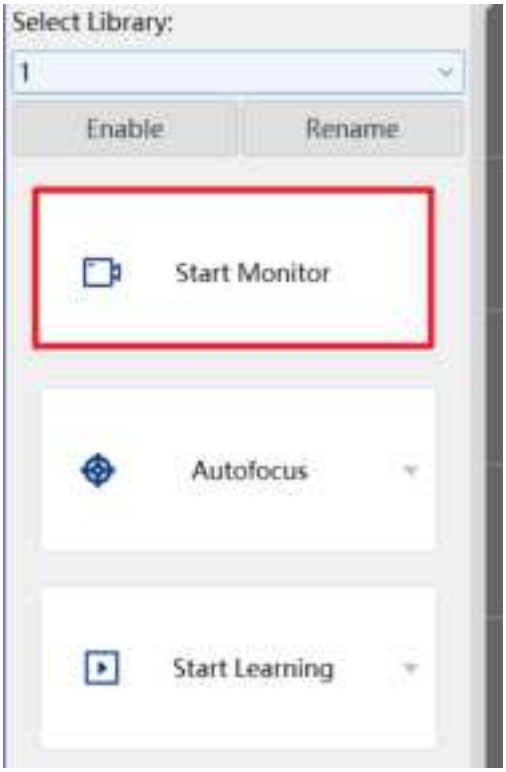
4.1 Select Library

Click "  " on the right to select the library. Tap "Rename" to quickly modify the library name. Click the "Enable" to enable or disable the library.

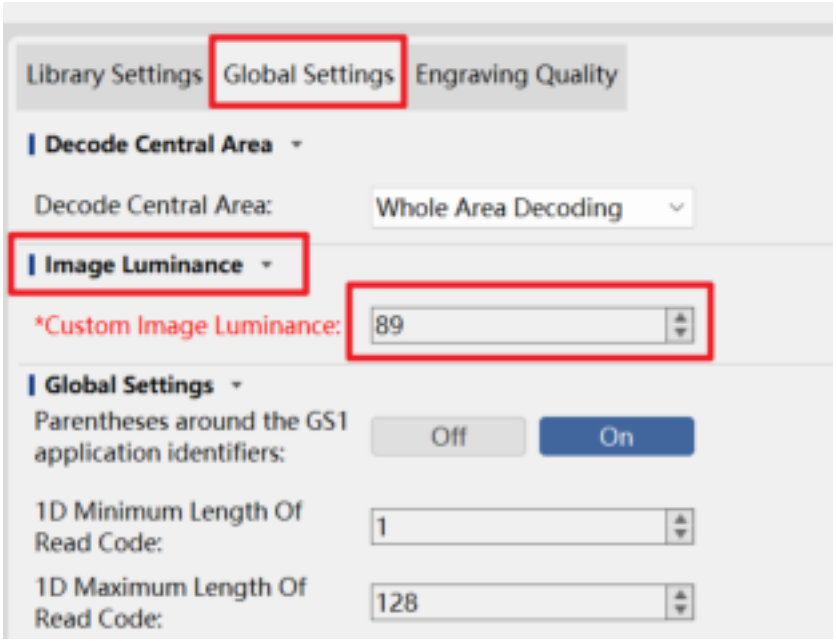


4.2 Start Monitor

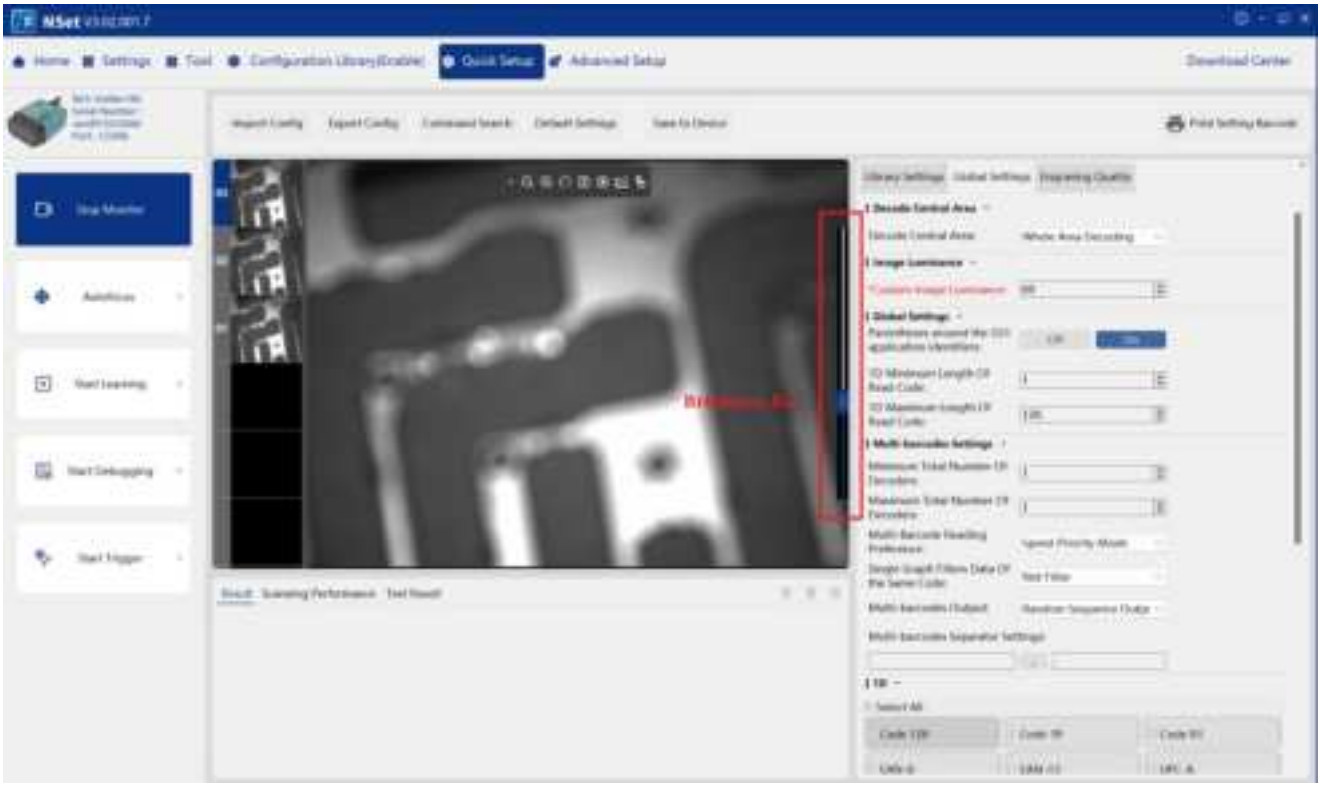
Click "Start Monitor" to activate the device to acquire images without decoding. The device will automatically optimize image brightness, which helps improve autofocus performance.



Brightness Optimization: Click "Quick Setup" > "Global Settings" > "Image Luminance".



Alternatively, after starting the monitor, you can adjust the brightness by dragging the brightness slider on the right side of the screen.



4.3 Autofocus

Click "Autofocus" to start auto focus.

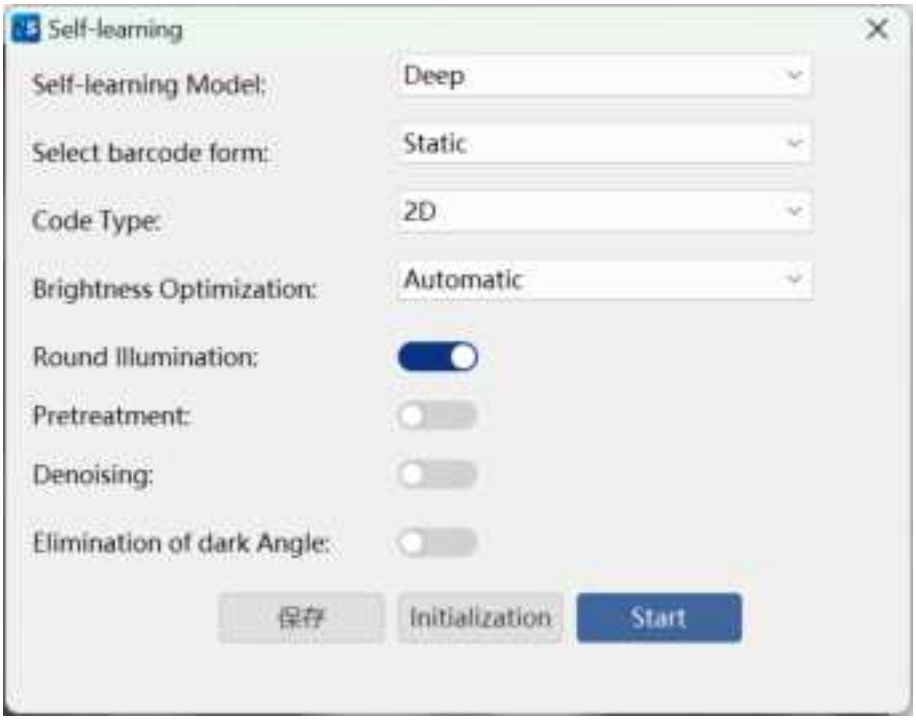
Click "▼" On the right side to set the focus modes:

- Focus Auto: The scanner automatically determines the position of the scanned object according to conditions on site, and then completes the focus.
- Focus Point: The mouse turns into a cross cursor, select or click anywhere in the image to focus on that position.
- Alternate Focus: When checked, the device will save different focus parameters to the configuration library with each autofocus operation.

4.4 Start Learning

Click "Start Learning" to start Auto-learning, click again to stop learning.

Click "▼" on the right side to set the learning parameters:



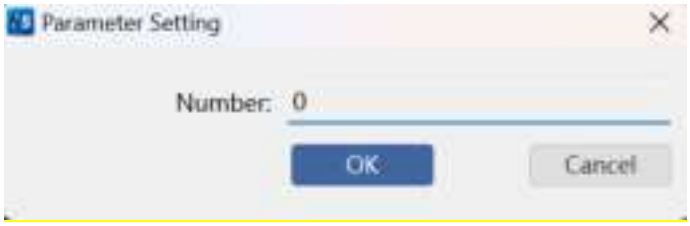
Parameter	Function
Self-learning Model	Deep mode by default.
Select Barcode Form	Select dynamic/static according to whether the barcode moves.
Code Type	Select the barcode type (1D or 2D) of the device learning object according to the actual requirements.
Brightness Optimization	Choose to automatically optimize the brightness or manually adjust the exposure time and gain.
Round Illumination	Choose whether to turn on the round illumination during the auto-learning process according to the actual requirements.
Denoising	Choose whether to enable denoising during the auto-learning process according to the actual requirements.
Elimination of Dark Angle	Choose whether to eliminate dark angle during the auto-learning process according to the actual requirements.

4.5 Start Debugging

Start Debugging: Only test the currently selected configuration library.

Click "Start Debugging" to start debugging, click again to stop debugging.

Click "▼" On the right side to set the debug parameters:



Parameter	Function
Number	Set the number of triggers. When the value is set to 0, the debug test runs continuously until manually stopped.

4.6 Start Trigger

Start Trigger: Test all enabled configuration libraries.

Click "Start Trigger" to start the barcode reading, click again to stop the barcode reading.

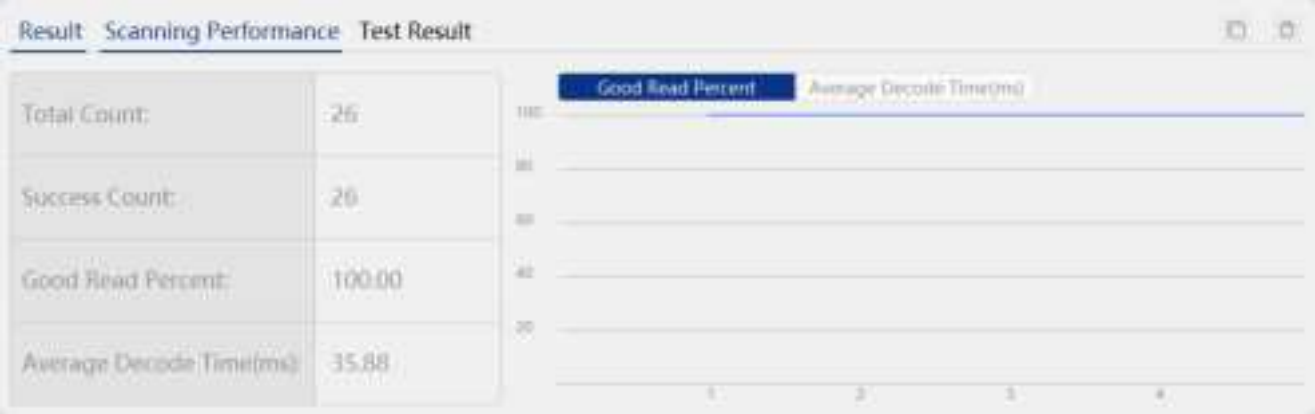
Click "▼" On the right side to set the trigger parameters:

Parameter	Function
Number	Set the number of times the device automatically triggers to read barcodes. When set to 0, the scanner will trigger continuously until manually stopped.

4.7 Result & Scanning Performance & Test Result

"Results": displays the decoded results in real time after each trigger.

"Scanning Performance": includes total count, success count, good read percent, average decode time, and line graphs of good read percent and average decode time.



"Test Result": includes barcode type, data, length, PPM, engraving quality, decoding time, configuration, and AI.

Note: Both **Start Trigger** and **Start Debugging** belong to the debug mode. The results are displayed in the **Test Result** section, while trigger actions initiated by the icon  or device button are shown in the **Result** section.

The statistical differences between debug mode and actual triggered decoding are as follows:

Parameter	Debug Mode	Actual Triggered Decoding
Test Content	Only tests the performance of the currently selected Library.	Tests the final application performance
Total Count	Number of decoded images	Number of trigger events
Success Count	Number of successfully decoded images	Number of successfully completed trigger events
Good Read Percent	Image decoding success rate (%)	Trigger event success rate (%)
Average Decode Time	Average image decoding time (ms)	Average decoding time for completed trigger event (ms)

4.8 Image Tool

Double-click the specific device in the device list on the home page, or click "Quick Setup", when the interface is USB CDC or Ethernet interface, you can view the real-time image read by the device through NSet, the tool at the top of the imaging window columns include:

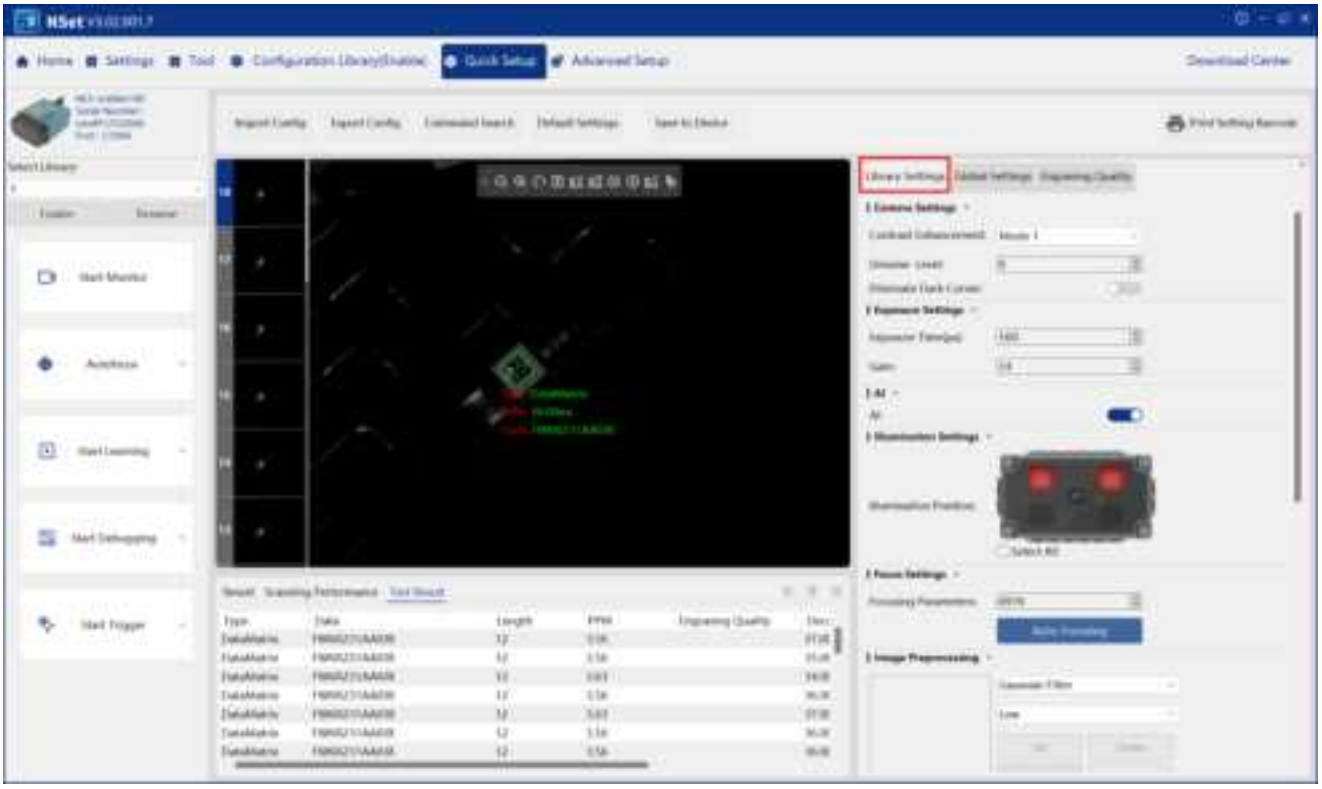
Icon	Function
	Zoom in on the current image
	Zoom out the current image.
	Restore the image size.
	Save image
	Save image to local file
	Set parameters such as image resolution and format.
	Autofocus

	Start/Stop real-time image view
	Obtain image
	Click to trigger device reading barcode.

Note: When the image format is set to raw, the image transmission time will increase, which may affect functions that rely on Ethernet transmission, such as networking.

4.9 Configuration Library

The Configuration Library allows you to set image and barcode parameters, including camera, exposure, illumination, and image preprocessing.



4.9.1 Camera Setting

Parameter	Description	Setting Range	Default Value
Contrast Enhancement	Enhances image contrast for better decoding accuracy	Standard/Mode1/Mode2/Mode3	Standard
Denoise Level	Reduces noise caused by high gain. Increase the level as gain increases.	0~7	0
Eliminate Dark Corner	Enables this function when decoding near the FOV boundaries.	0: Disable 1: Enable	0

4.9.2 Exposure Setting

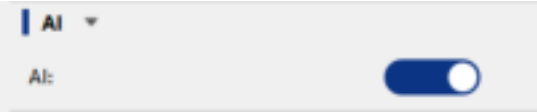
NSet can adjust the exposure time according to the scanned object. When the exposure time is set too short, the scanning object that moves faster can be read, but the field of view becomes dark due to insufficient exposure, which will make decoding difficult; and when the exposure time is too long, the captured barcode image will be smearing and blurring. Please adjust the exposure time value appropriately according to the production line speed.

If the on-site light source and illumination are unstable, the exposure and gain parameter values can be set to an interval, and the scanner will adjust within this interval every time the barcode is scanned (but it will affect the decoding performance).

Parameter	Description	Setting Range	Default Value
Exposure Time((μs)	Set the exposure duration	60~60000	1000
Gain Value	Set the gain value	1~128	1

4.9.3 AI

Enable or disable AI function per actual need.



4.9.4 Illumination Position

Adjust the position of illumination lights.

Parameter	Description	Setting Range	Default Value
Illumination Position	Manually turn each illumination light on or off.		All Selected

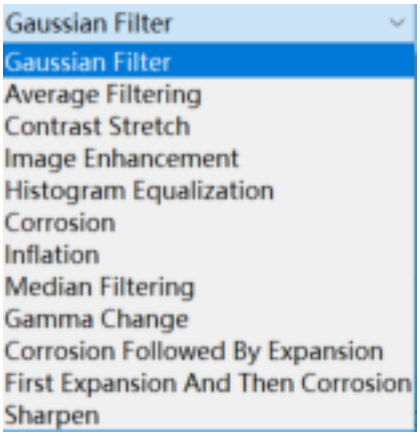
4.9.5 Focus Setting

You can adjust the focal distance in real time by entering focusing parameters, or click the Auto-Focusing button to enable automatic focusing.

4.9.6 Image Preprocessing

During the reading process of the device, the image quality directly affects the decoding accuracy of the recognition algorithm, thereby affecting the decoding speed and bit error rate. Therefore, in order to further improve the speed and reliability of decoding, it is necessary to effectively preprocess the image. Common barcode problems include: Noise in the barcode area, low-contrast code, and barcode printing diffusion.

Currently, the preprocessing that the device can support are as shown below:



According to the requirements, click "Add" And then select the required preprocessing function and preprocessing intensity. Multiple preprocessing modes can be added at the same time, and they will be superimposed and sorted in order from top to bottom.

Image Preprocessing Example

Type	Barcode features	Before processing	After processing	Description
Contrast Stretch	Low contrast and no noise			Useful for low contrast and no bright spots in the image.
Histogram Equalization	Low contrast +lots of noise			Useful when contrast is low and there are bright spots in the image.
Corrosion	Black print spread			After corrosion treatment, the white blocks in the barcode become wider, and the black blocks are eroded and narrowed.
Inflation	White print spread			Contrary to erosion, after the expansion process, the white block part in the barcode is narrowed, and the black block is expanded and widened.
Gaussian Filter	Code area is noisy			The legend is a simulated qr barcode with 10% salt and pepper noise. The gaussian filter is softer than the mean filter, and the edge is better preserved.
Average Filtering	Code area is noisy			The denoising effect is weak, and the advantage is that it is fast and can be used when the noise is not serious.

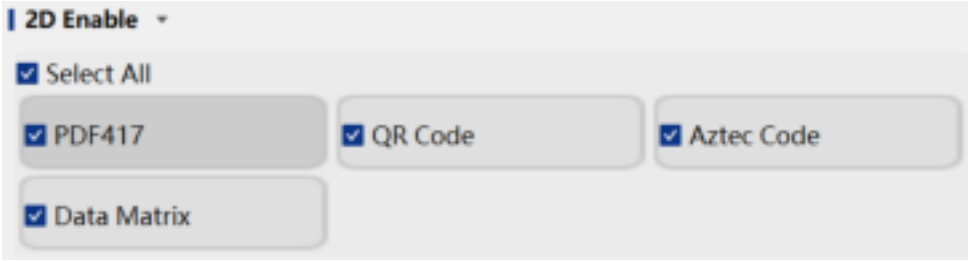
Median filtering	Code area is noisy			Median filtering is slow and very effective at smoothing out impulse noise, while protecting sharp edges in the image, and is used in very noisy situations.
Image Enhancement	Low contrast			Useful for underexposed, low-contrast images without bright spots.
Gamma Change	Low brightness			Useful for images with low contrast or extreme lighting conditions (too bright or too dark).
Corrosion Followed By Expansion	Small holes and gaps within black regions			Used to fill small holes and gaps within black regions.
First Expansion And Then Corrosion	Small holes and gaps within white regions			Used to fill small holes and gaps within white regions.
Sharpen	Blurred images			Used to enhance edges and details in blurred images.

4.9.7 Enable 1D Symbology

Enable 1D symbology as per actual need

4.9.8 Enable 2D Symbology

Enable 2D symbology as per actual need.



4.10 Global Setting

4.10.1 Decode Central Area

Two decoding modes are available: Whole Area Decoding and Multiple ROI Decoding.

Parameter	Description	Setting Range	Default Value
Decode Central Area	Whether to turn on Multiple ROI Decoding	Whole Area Decoding Multiple ROI Decoding	Whole Area Decoding

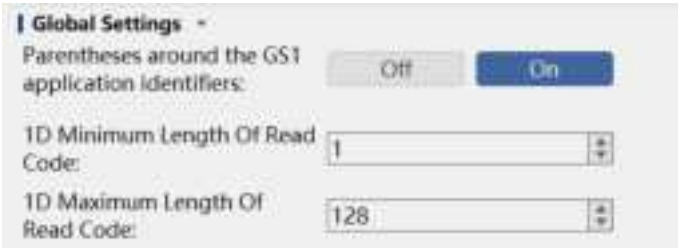
4.10.2 Image luminance

Image luminance is the key parameter of scanner configuration, and proper image gray value will ensure successful decoding. Using a live image or acquiring an image will help you adjust the image luminance value. For test, you can directly adjust the image luminance in order to simplify the setup. (It is not recommended to set image luminance in mobile scanning)

Parameter	Description	Setting Range	Default Value
Image luminance	Adjusts image luminance when monitor mode is enabled.	1-255	100

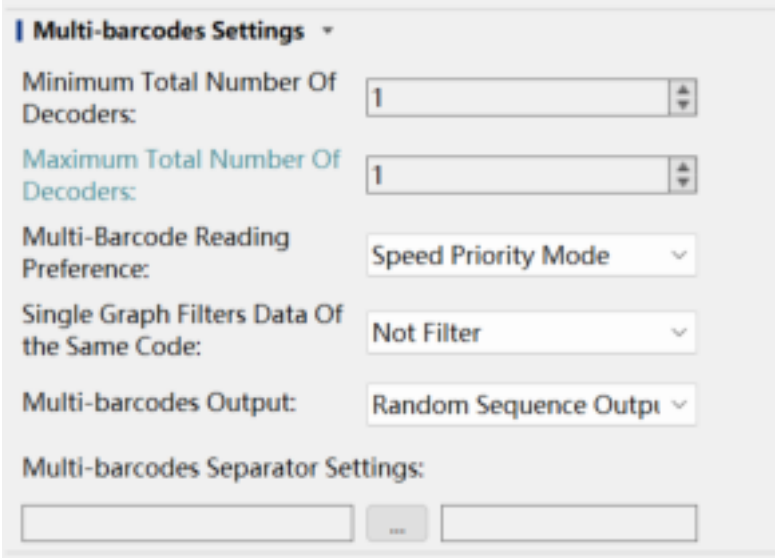
4.10.3 Global Settings

Global settings can configure parameters such as enabling parentheses around GS1 application identifiers, and setting the minimum and maximum length of 1D barcode decoding.



4.10.4 Multi-barcode Settings

The multi-barcode settings allow you to configure decoding constraints and the output sequence for multiple barcodes. The output sequence also applies in synchronized mode during networked operation.



Minimum and maximum Total Number of Decoded

The output decoding quantity is randomly output according to the minimum to maximum quantity (but the lower limit quantity is according to the minimum setup value. For example, the minimum setup is 2, the maximum setup is 6, and one decoding can output 2-6 random barcodes, but at least 2 will be output).

Scan Preference

Two modes are available: Speed Priority and Quantity Priority.

Single Graph Filters Data of the Same Code

Filter data for the same barcode in the same image.

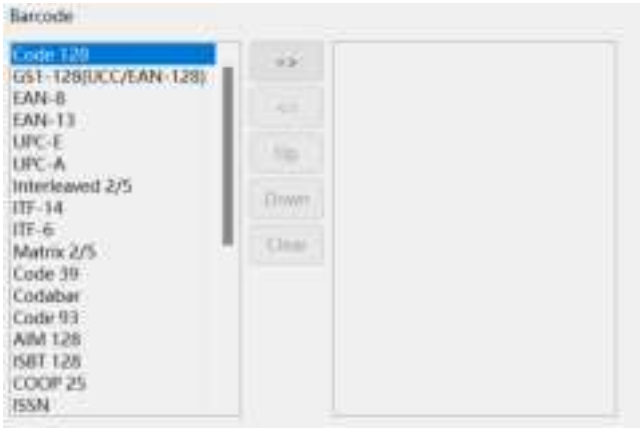
Multi-barcodes Output

- **Random Sequence Output**
  - Output barcode information in random order.

● Fixed Sequence Output Mode

Customize the sorting and output barcode information according to the characteristics of symbology type, vertical direction, horizontal direction, etc., and adjust the output priority by moving up/down.

Click "=> / <=" to add/remove symbology types and adjust output priority by moving up/down.



Multi-barcodes Separator Settings

Customize the separator between multiple barcodes.

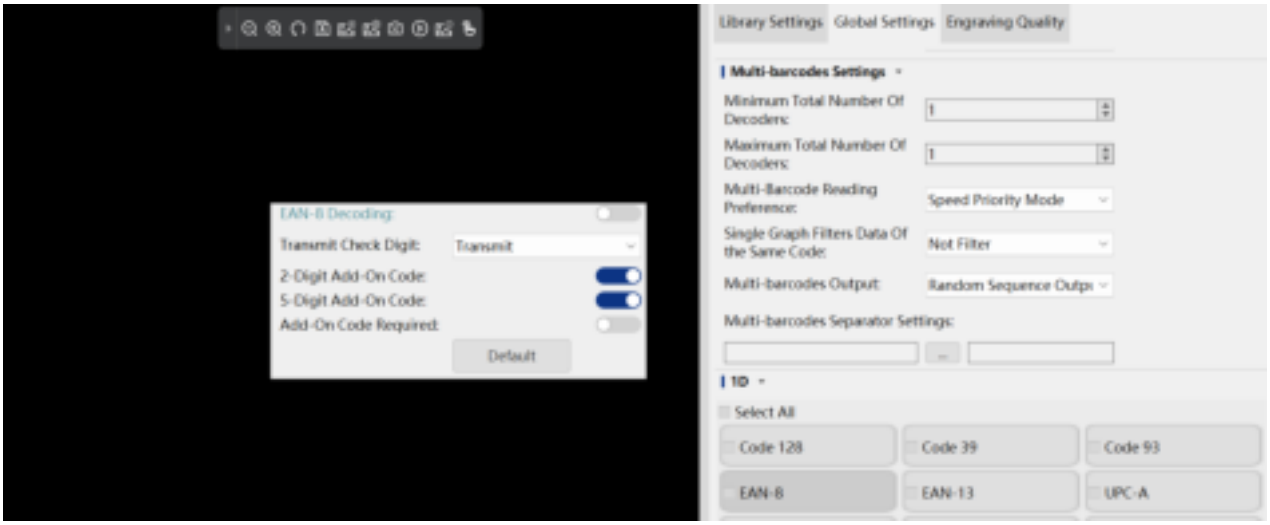
Example

Set the barcode separator to "--", the output result is:

A01--B0002--C03--D001--E001--2112345678917。

4.10.5 1D

Configure parameters for 1D barcodes, such as transmit the check digit and add-on code required. (Not limited to a specific configuration library.)



4.10.6 2D

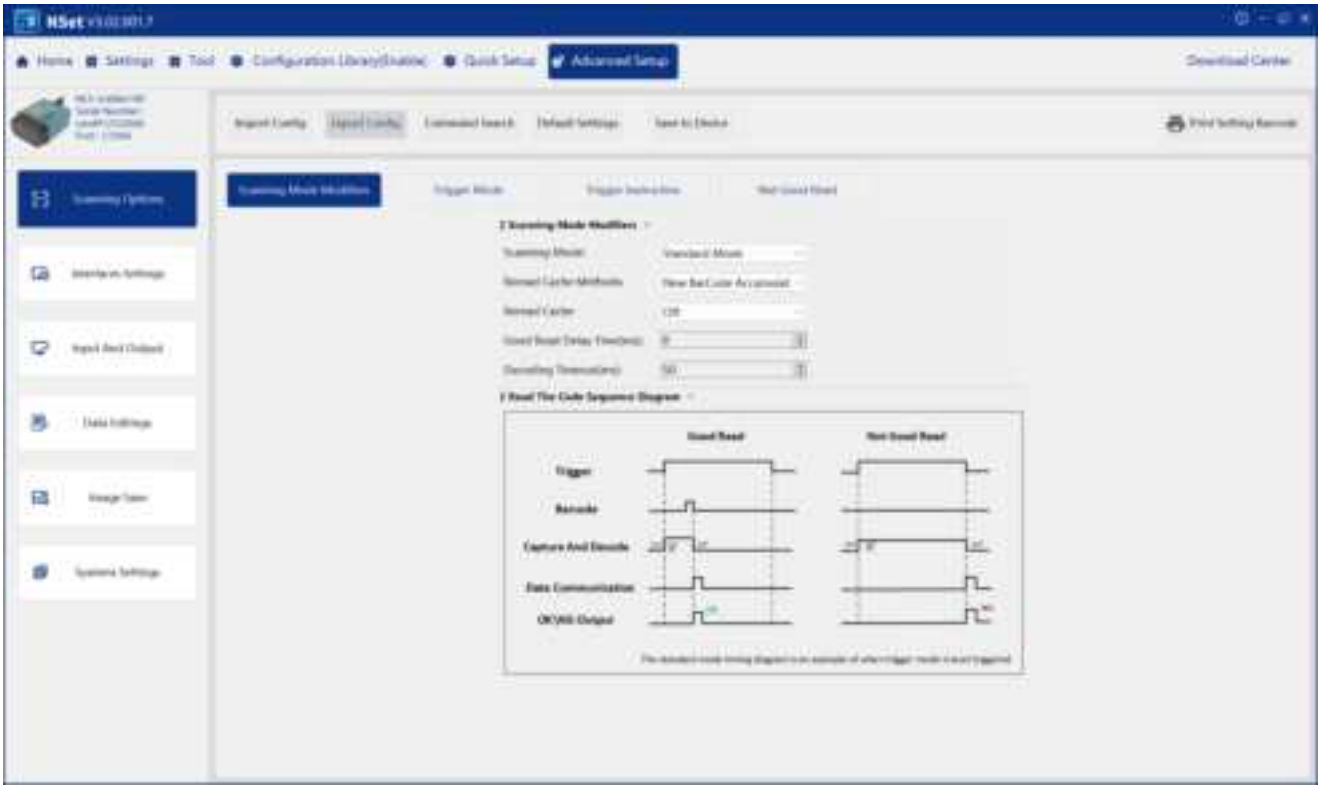
Configure parameters for 2D barcodes, such as minimum length, maximum length, and ECI Output.

4.11 Engraving Quality

Engraving Quality: displays the engraving quality grade in the current decoding state.

5 Advanced Setup

Advanced Setup include: Scanning Options, Interfaces Settings, Input and Output, Data Editings, Image Save, System Settings



5.1 Scanning Options

The Scanning Options include various scan-related configurations, such as scan parameters, trigger modes, and other related settings.

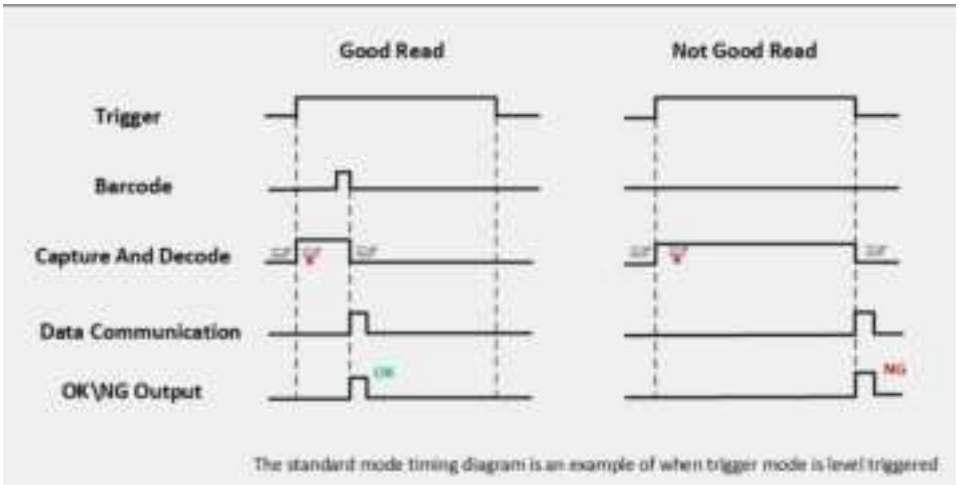
5.1.1 Scanning Settings

The scanner provides three scanning modes, as described below:

Number	Function	Description
1	Level Mode	Reads one barcode during each trigger event
2	Continuous Mode	Continuously reads barcodes while the trigger is held
3	Burst Mode	Captures multiple images per trigger and decodes one barcode

Level Mode

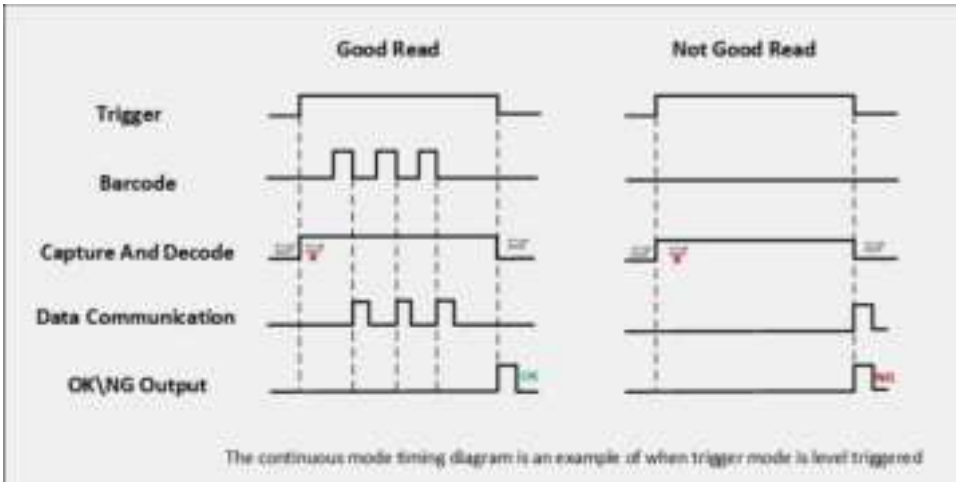
A trigger pull activates a decode session. The decode session continues until a barcode is decoded or you release the trigger.



Parameter	Description	Setting Range	Default Value
Decoding Timeout(ms)	Defines the maximum duration that the barcode scanner will attempt to read and parse the scanned object, after which stop this decoding.(same for Continuous Mode and Burst Mode)	1~3000	500

Continuous Mode

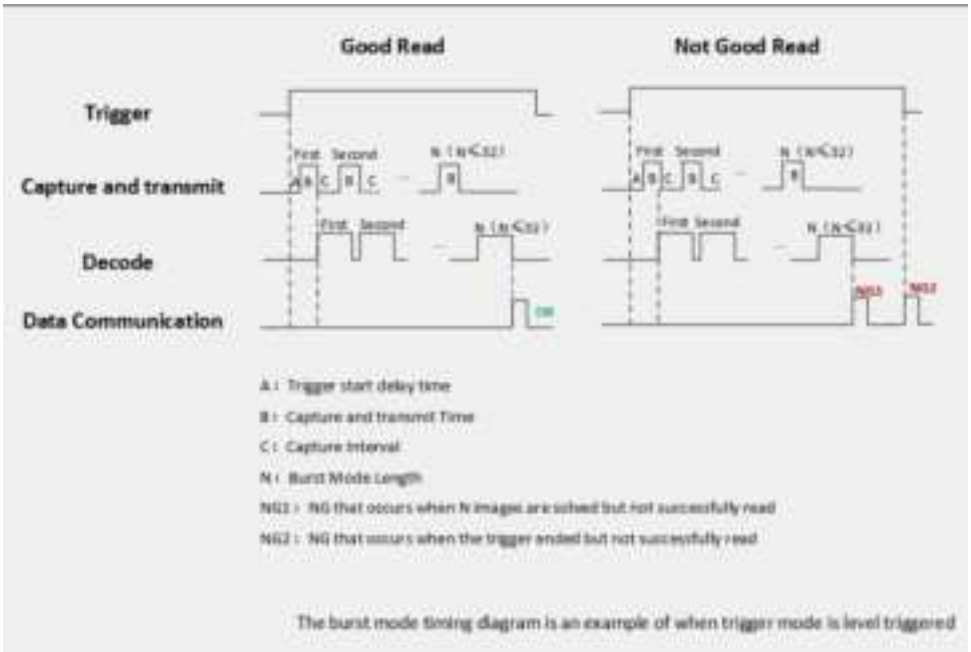
Press the scan trigger to continuously read barcodes. In this mode, the reread delay can be used to prevent the same barcode from being read multiple times.



Parameter	Description	Setting Range	Default Value
Reread Delay(ms)	The barcode scanner reads the scanned object according to the set time interval, and the same barcode will not be read repeatedly during the time interval.	0~3600000	0
Good Read Delay Time(ms)	The length of time to pause the barcode reading after successful barcode reading	0~3600000	0

Burst Mode

This configuration is the exclusive mode for the fixed barcode scanners. The reading is started by pressing a button or an external trigger signal. After starting, the image is decoded while the image is acquired. After the maximum buffer is 32 images, the image is stopped. At this time, only decoding is performed, and the exit condition is that the decoding is successful, or finish 32 cached pictures, or press the button again to start reading. Conditions for entering the reading state: button press or external trigger signal. Conditions for exiting the reading state: decoding is successful or reading barcode times out, or 32 cached pictures are decoded. After exiting the reading state: wait for a key press or external trigger input. The burst mode is applied to the fast moving usage scenario of a single barcode. According to the moving speed of the scene, the length of the barcode, the interval time, etc., it can be set in a targeted manner and get the optimal setting.



The burst mode timing diagram is an example of when trigger mode is level triggered

Parameter	Description	Setting Range	Default Value
Burst Mode Trigger Delay Time(ms)	After the trigger signal is valid, it will enter the reading state after a delay for a period of time	0~1000	0
Burst Mode Length	This parameter reflects the barcode moving speed, the default is 8, and the calculation formula is: Burst Mode Length = Scanning Range (mm)/Barcode Movement Speed (mm/s) * 1000/20	1-32	8

Output logic differences between three scanning modes

Scanning Modes	OK	NG
Level Mode	Output immediately after successful read	Output after trigger release
Continuous Mode	Output after trigger release	
Burst Mode	Output immediately after successful read	Output after trigger release

Reread cache method

Number	Parameter	Description
1	New Barcode Accumulate	New scanned barcodes are output and accumulated in the reread cache
2	New Barcode Coverage	New scanned barcodes are output and overwrite the previous barcode in the reread cache

Reread cache modes

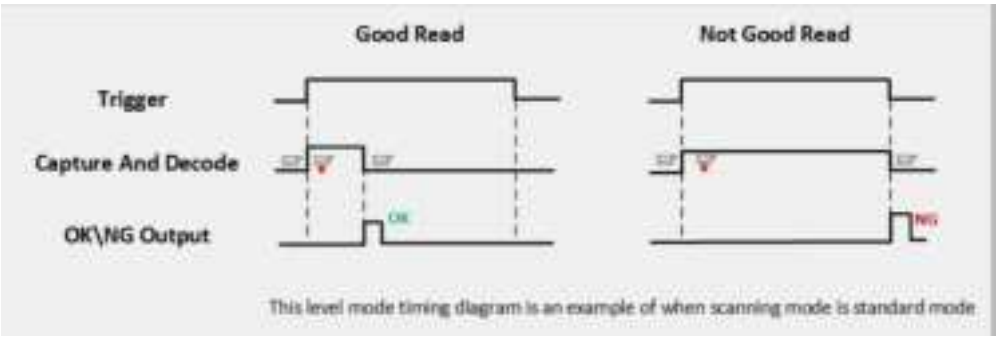
Number	Parameter	Description
1	Off	Disable reread delay function
2	Cache within Reread Delay Time	Barcodes scanned during the reread delay period are cached and will not be read again within this period. Note: The reread delay time is unaffected by the trigger OFF time.
3	Cache during Read	Barcodes scanned during an active trigger session are cached and will not be read again until the trigger is released (end of session).

5.1.2 Trigger Mode

The trigger mode can be configured based on the input signal and application requirements.

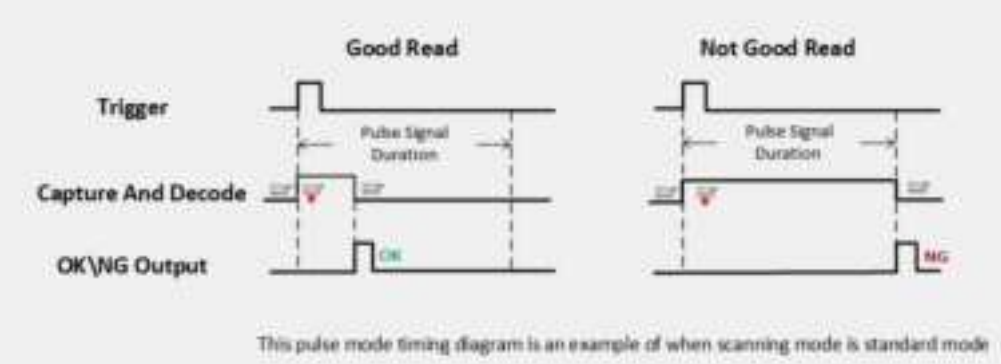
Trigger Mode	Description
Level Trigger Mode	When the trigger input is ON, the LED illumination remains active, and the scanner will start reading. If a barcode is successfully decoded, the illumination turns off, and the data is transmitted
Pulse Trigger Mode	The scanner detects the rising edge of the trigger input signal. The LED illumination turns on, and the scanner performs a read attempt for a specified duration. If a barcode is successfully decoded, the illumination turns off and the data is transmitted.

Level Trigger Mode



Parameter	Description	Setting Range	Default Value
Trigger Startup Delay Time(ms)	Specifies the time delay between receiving a trigger signal and the scanner starting to decode (also applies to pulse trigger mode).	0~60000ms	0ms
Trigger Stop Delay Time(ms)	Specifies the time delay between receiving a trigger release signal and the scanner stopping decoding.	0~5000ms	0ms

Pulse Trigger Mode



Parameter	Description	Setting Range	Default Value
Pulse Signal Duration(ms)	Defines the duration of scanner operation after detecting a pulse trigger signal.	30~360000ms	3000ms

The differences between Level Trigger Mode and Pulse Trigger Mode are as follows:

Trigger Mode	Trigger Start	Trigger Stop
Level Trigger Mode	Activated upon level signal detection	Deactivated when level signal ends
Pulse Trigger Mode	Activated upon pulse signal detection	Deactivated after preset pulse signal duration expires

5.1.3 Trigger Instruction

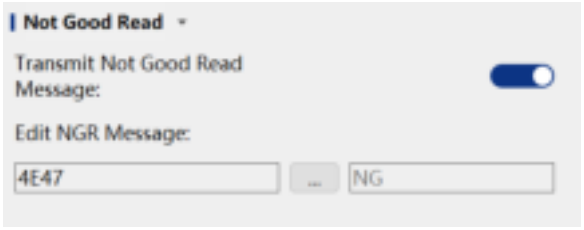
Serial Port Trigger Instruction: Click to enable or disable the serial port trigger instruction function. The command trigger operates similarly to pulse trigger mode.

Edit Start Scanning Instruction: The default start scanning instruction is 015405, and the default stop scanning instruction is 015004. Users can customize these instructions as needed.

5.1.4 Not Good Read

The barcode reading unsuccessful message supports 1 to 32 characters, the value range of the character is 0 to 0xff, and the default is NG.

Parameter	Description	Setting Range	Default Value
Transmit Not Good Read Message	Enable/Disable transmit not good read message	1：Enable 0：Disable	0



5.2 Interface Settings

5.2.1 RS-232 Settings

When the barcode scanner and the host are connected by a serial cable, both parties need to set the same communication parameters to ensure the normal communication. The parameters to be set include the communication baud rate, check character, stop bit, data bit.

Baud Rate

The baud rate is the number of bits transmitted per second in serial data communication. The baud rate used by the barcode scanner and the data receiving host must be consistent to ensure the accuracy of data transmission.

RS-232 Check

The barcode scanner can choose different parity check character types in the process of using the serial port transmission, but it must be consistent with the parity check character type of the host.

Select odd check: If the number of "1" In the transmitted data is odd, the check character is 0.

Select even check: If the number of "1" In the transmitted data is even, the check character is 0.

Select no check: No parity characters are sent.

RS-232 Stop Bits

The stop bit is located at the last part of each byte transmitted, and is used to mark the completion of the transmission of this byte to start receiving the next byte of data.

1 stop bit is set by default. If you need to stop for a long time, you can set 2 stop bits.

RS-232 Data Bits

You can choose to transmit 7 or 8 bits of data, please make sure that the data bits of the barcode scanner are consistent with those of the data receiving host.

5.2.2 Keyboard Modifiers

Using USB connection, the scanner can be set to hid keyboard input mode. In this mode, the barcode scanner will become a virtual keyboard, and the data receiving host will accept the input of this virtual keyboard just like the real keyboard input. After the barcode scanner decodes and obtains the data, the sending process is to hit each key corresponding to the data in the virtual keyboard.

If the input box of the host can accept keyboard input, the barcode scanner can directly input the decoded data into the input box of the host without any other auxiliary programs using this communication method.

Keyboard Country

The keyboard keys and symbols corresponding to different national languages are not the same. Therefore, the barcode scanner can be virtualized into keyboard systems of different countries as required, and the default is the keyboard of the US standard.

Beep on Unknown Character

If the "Emulate ALT +Keypad" function is enabled, this function is invalid. Due to differences in national keyboard layouts, some characters contained in barcode data may not be available on the selected keyboard. For this, the scanner will skip this unknown character and continue processing the next character. It is recommended to turn on the beep on unknown character.

Emulate ALT +Keypad

When "Emulate ALT +Keypad" Is enabled, the characters will be sent through the numeric keypad, ignoring the national keyboard layout setup. This mode also needs to set Output Unicode and Code Page. Output Unicode controls whether the encoding method of input to the host is Unicode or Code Page, and Code Page determines the target language.

This method of sending ensures that any character can be transmitted smoothly, but is slower because of the large number of keystrokes that need to be simulated for each character transmitted.

Assuming that the current Code Page selection of the barcode scanner is Code Page 1252 (Latin, Western Europe), the barcode scanner can read the barcode with the content "AÐF" (the decimal value in Code Page 1252 is 65/208/70).

If the Emulate ALT +Keypad is set to "On" And the Output Unicode is set to "Off", the scanner emulates the keyboard operation as follows:

Input "A" - press the alt key, simultaneously press the numeric keypad keys 0, 6, 5 in sequence, and release the alt key;

Enter "Đ" - press the alt key, simultaneously press the numeric keypad keys 2, 0, 8 in sequence, and release the alt key;

Enter "F" - press the alt key, simultaneously press the numeric keypad keys 0, 7, 0 in sequence, and release the alt key.

Control Characters Output

If the "Emulate ALT +Keypad" function has been enabled, this function is invalid.

The ASCII characters between 0x00~0x1f can be escaped into a certain control function key. For the corresponding relationship between the specific ASCII value and the control function key, please refer to Appendix. Default is off.

When other hid keyboard related setup of the barcode scanner are set to the default values, when this setup is set to control character output "Control + ASCII mode", the reading data is "A<HT>F (HT is an invisible character, which is not displayed on the software terminal)" (the hexadecimal value is 0x41/0x09/0x46), the virtual keyboard operation of the barcode scanner is as follows:

Input "A" - press button A;

Enter "Ctrl I" - since the data of 0x09 corresponds to the control function key "I", the virtual keyboard will hold down ctrl, then press the I key, and then release the ctrl key and the I key at the same time;

Enter "F" - press key F.

Since "Ctrl I" Corresponds to the function of converting characters to italics in some word processing software, you may see the normal character "A" And the italic "F" After completing the above operation.

Inter-Keystroke Delay

This parameter specifies the delay between simulated keypresses. Default is no delay.

Long delay: 40ms;

Short delay: 20ms.

Caps Lock

If the "Emulate ALT +Keypad" Or "Case conversion" function is enabled, this function is invalid.

When on, uppercase and lowercase characters contained in barcode data can be reversed. This inversion occurs regardless of the state of the caps lock key on the host keyboard. Default is off.

After this function is turned on, the barcode scanner reads the barcode with the data "AbC", and the host will get "aBc".

Convert Case

If the "Emulate ALT +Keypad" function has been enabled, this function is invalid.

Convert All to Upper Case: Convert all to uppercase letters regardless of whether the letters in the barcode data are uppercase or lowercase.

Convert All to Lower Case: No matter whether the letters in the barcode data are uppercase or lowercase, all the letters are converted to lowercase letters.

Numeric Character Use Numeric Keypad

If the "Emulate ALT +Keypad" function has been enabled, this function is invalid.

If this function is not enabled, all outputs will be output according to the corresponding key value of the main keyboard.

After this function is enabled, if the barcode data read by the barcode scanner contains numbers "0~9", the virtual keyboard will output the key value corresponding to the numeric keypad.

The Num Lock in the upper left corner of the numeric keypad controls whether its key value is a number or a function key. The virtual keyboard does not control the Num Lock state independently, but is consistent with the Num Lock state of the actual keyboard of the host computer. Therefore, if the actual keyboard of the host computer turns off Num Lock (the Num Lock light is off), after the virtual numeric keypad of the barcode scanner, the output is function keys instead of numbers. Therefore, be sure to confirm the Num Lock status of the host before using this function.

'+', '-', '*', '/' Use Numeric Keypad

If the "Emulate ALT +Keypad" function has been enabled, this function is invalid. That is, '+', '-', '*', '/' on the numeric keypad.

USB Polling Rate

The lower the value of the keyboard polling speed setup, the faster the scanner can send characters to the host. If the host discards characters, it is recommended to increase the polling speed setup.

Output Unicode

Different application software also has requirements for the received character encoding. For example, MS office's word uses Unicode encoding, and Unicode output needs to be set to "On"; while MS office's excel or notepad uses code page encoding, you need to set the Unicode output to "On". Set Unicode output to "Off".

The "Unicode output" Function only takes effect when the "Emulate ALT +Keypad" Function is turned on. Default is off.

Code Page

If the "Emulate ALT +Keypad" function has been enabled, this function is invalid. The barcode page defines the mapping of character codes to characters, default is code page1252 (Latin, Western European).

5.2.3 Output Channel Settings

A total of nine barcode output channels: RS-232, USB, TCP Client, TCP Server, UDP, Modbus TCP Client, Profinet, EtherNet/IP, MELSEC/SLMP.

Among them, the factory cables of NVF and Soldier series products include USB, serial port and Ethernet. Therefore, such devices can connect to multiple interface types at the same time to output scanning results, or select multiple output channels at the same time.

USB

There are three types of USB output channels: USB HID Keyboard Output, HID POS barcode Output and USB CDC COM Output. Choose a USB output method to output barcodes according to actual needs.

If a certain USB interface type has been selected in the interface type, only the corresponding interface type can be selected in the USB output channel to output barcode data, and other USB interface types will not be able to output barcode data (only for the interface type selected USB type case).

For example, if USB CDC COM port is selected as the Interface type, but USB HID Keyboard Output is selected in the USB output channel, the barcode data cannot be output.

TCP Client

The remote IP address is the IP address of the host computer TCP server used to receive barcode data, and the remote port number is the port number monitored by the host computer TCP server.

TCP Server

The TCP server is provided to the host computer to connect to the TCP client and receive barcode data. The IP address of the TCP server is the IP address of the device. If it needs to be changed, it needs to be modified on the "Home: Device" Page. The port number of the TCP server is the port number monitored by the TCP server.

UDP

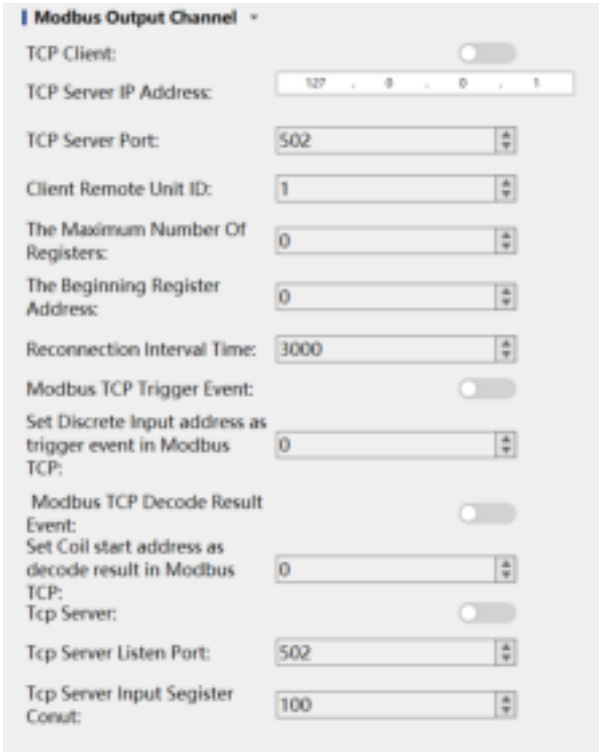
The UDP remote IP address is the IP address of the host computer UDP server used to receive barcode data, and the remote port number is the port number monitored by the host computer UDP server

Industrial Ethernet Protocol

Three options are available: Disable, Enable Profinet, and Enable EtherNet/IP.

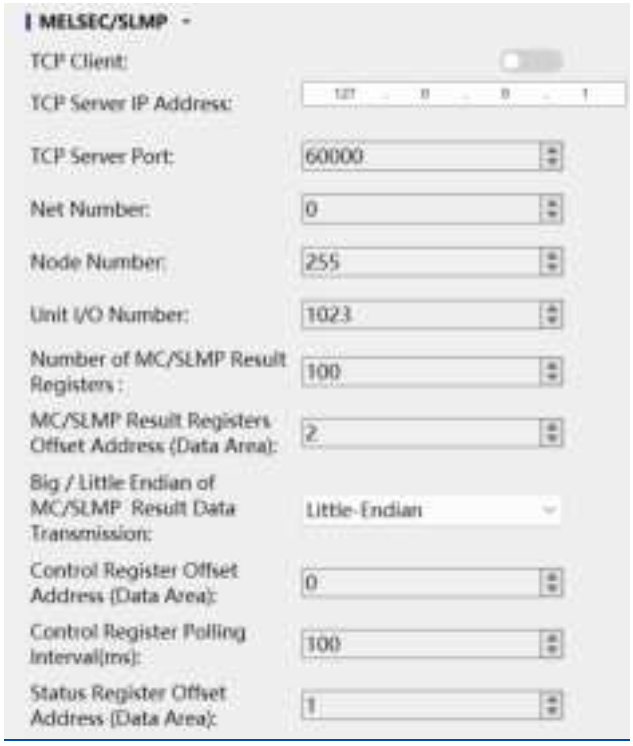
Modbus Output Channel

To enable the Modbus TCP client, you need to configure options such as the TCP Server IP Address, TCP Server Port, Client Remote Unit ID, the Maximum Number of Registers, the Beginning Register Address, and the Reconnection Interval Time.



MELSEC/SLMP

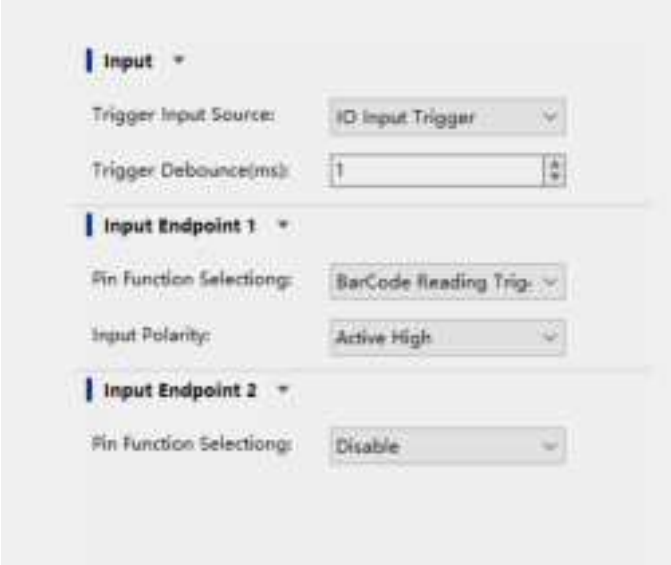
To enable the MELSEC/SLMP protocol, you need to configure the TCP server IP address, TCP server port, Net number, Node number, and other related parameters as required.



5.3 Input and Output

The Input/output section is used to configure functions related to I/O signals, commands, and other relevant features.

5.3.1 Input



Trigger Input Source: used to select the source of the trigger signal. Available options include IO input trigger, serial instruction trigger, and simulate input trigger.

Trigger Debounce (ms): defines the time interval to debounce the trigger signal. Set this value according to the waveform characteristics of the external input signal.

Input Endpoint 1 & Input Endpoint 2: two input terminals are available on the scanner, and set the configurations per actual needs.

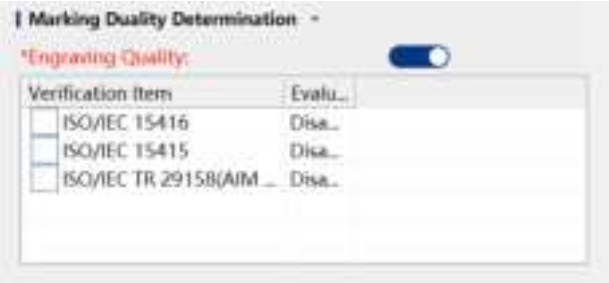
Parameter	Function
Disable	Disable the function of this terminal
Barcode Reading Trigger	Starts or stops barcode reading based on the input signal.

5.3.2 Output

Two output terminals are available on the scanner, and set the configurations per actual needs.

Parameters	Functions
Disable	Disables the function of this terminal.
Good Read Indicator	Controls the output terminal after a successful read
Bad Read Indicator	Controls the output terminal after a failed read
Auto-learning Success Indicator	Controls the output terminal after a successful Auto-learning
Auto-learning Failure Indicator	Controls the output terminal after a failed Auto-learning verification.
Marking Verification Success Indicator	Verifies if the marking quality meets the preset threshold

The Marking Verification Success Indicator requires the Engraving Quality function to be enabled in the data editing interface.



5.3.3 IO Serial Trigger

The input/output terminals can also be triggered directly by external commands sent via serial communication. This feature is commonly used when the client system verifies the barcode and sends a trigger command to activate the scanner's output terminals, enabling subsequent processes.

5.4 Data Editings

After the barcode scanner decodes successfully, it obtains a string of data, which can be numbers, English, symbols, etc., and can also be Chinese characters for QR code. This string of data is the data information contained in the barcode. In practical applications, we may not only need the data information of the barcode, or the data information contained in the barcode cannot meet your needs. For example, you may want to know which type of barcode the obtained string of data information comes from, or what day the barcode information was scanned, or you hope that after scanning a barcode, the text recording the barcode can automatically wrap and enter, which may not be included in the data information of the barcode.

Soldier180 supports advanced data editing mode, set the required output information by writing a script, and click "  " to get the script help document.

5.4.1 Data Editings

Data Output Information

Two modes are available: Normal mode and Script mode. Supports JavaScript for custom data editing (add/delete/modify).

Prefix

Adding these contents during barcode production will inevitably increase the length of the barcode and lack flexibility, which is not an advocated practice. At this point, we thought of adding some content before or after the data information of the barcode, and these added content can be changed in real time according to the needs, and you can choose to add or shield it. This is the prefix and suffix of the barcode data information. This method meets the requirements without modifying the content of the barcode information.

Custom Prefix

A custom prefix adds a user-defined string before decoding the information.

For example, it is allowed to add a custom prefix and set the prefix to the string "Ab". After reading the barcode with the data "123", the scanner adds the "Ab" String before the "123" String, and the host receives the "Ab123" String.

First open "Custom Prefix", then read the hexadecimal value of each byte in the prefix string to be set in sequence, and finally read "Save setup" to complete the custom prefix setup.

The total length of the custom prefix string cannot exceed 10 characters, and the character value range is 0x00~0xff.

Suffix

Custom Suffix

The custom suffix is to add a user-defined string after decoding the information.

For example, it is allowed to add a custom suffix and set the suffix to the string "Ab". After reading the barcode with the data of "123", the scanner adds the "Ab" String after the "123" String, and the host receives the "123Ab"" String.

First open "Custom suffix", then read the hexadecimal value of each byte in the suffix string to be set in sequence, and finally read "Save setup" to complete the custom suffix setup.

The total length of the custom suffix string cannot exceed 10 characters. The character value range is 0x00~0xff.

5.4.2 Barcode Additional Information Settings

Select the additional information to be included in the output barcode data, including: Barcode Data, Barcode Number, Barcode Coordinates, Decoded Time, etc.

The data separator can be customized between multiple content items, the default is " , ", as shown below:

5123827: Barcode Data;

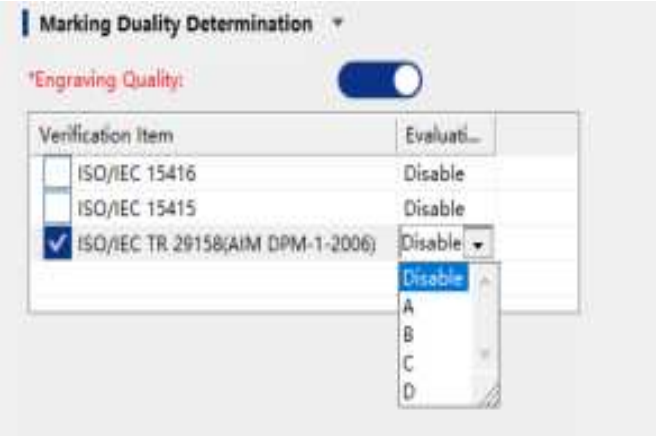
2: Barcode Number;

(1068,659): Barcode Coordinates;

133ms: decoded Time.

5.4.3 Marking Quality Determination

The marking quality determination can add barcode grade information to the data and set an evaluation threshold. You can choose whether to output the evaluation result in the output terminal function settings.



5.5 Image Save

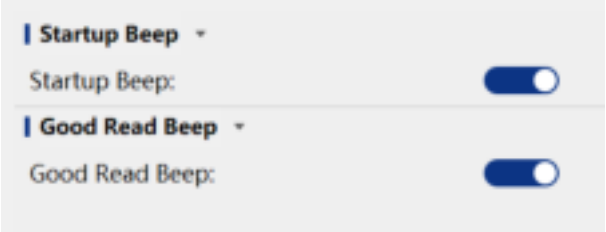
5.5.1 Image Save

Save the image to the device RAM; or the specify ftp server, you need to fill in the ftp address, port number, user name and password.

5.6 System Settings

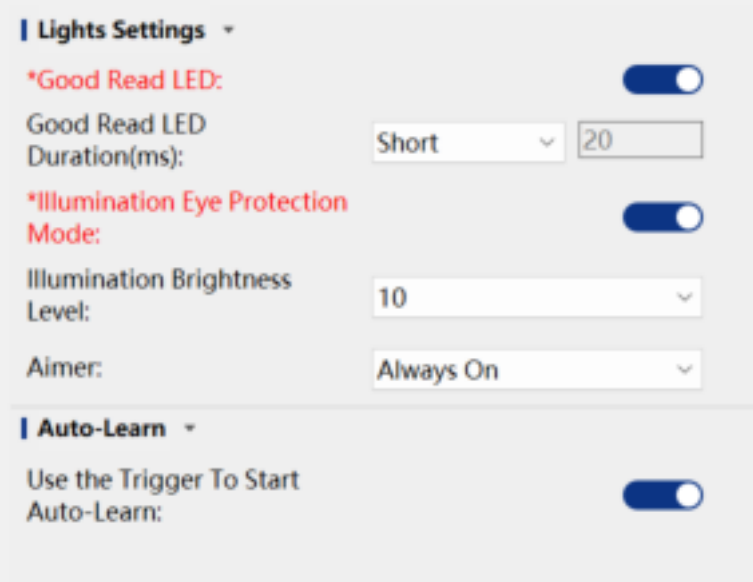
5.6.1 Beep

Enable or disable the Startup Beep or Good Read Beep.



5.6.2 Hardware Settings

Parameters of the Good Read LED, Illumination Eye Protection Mode, aimer, and auto-learn can be configured based on actual requirements.



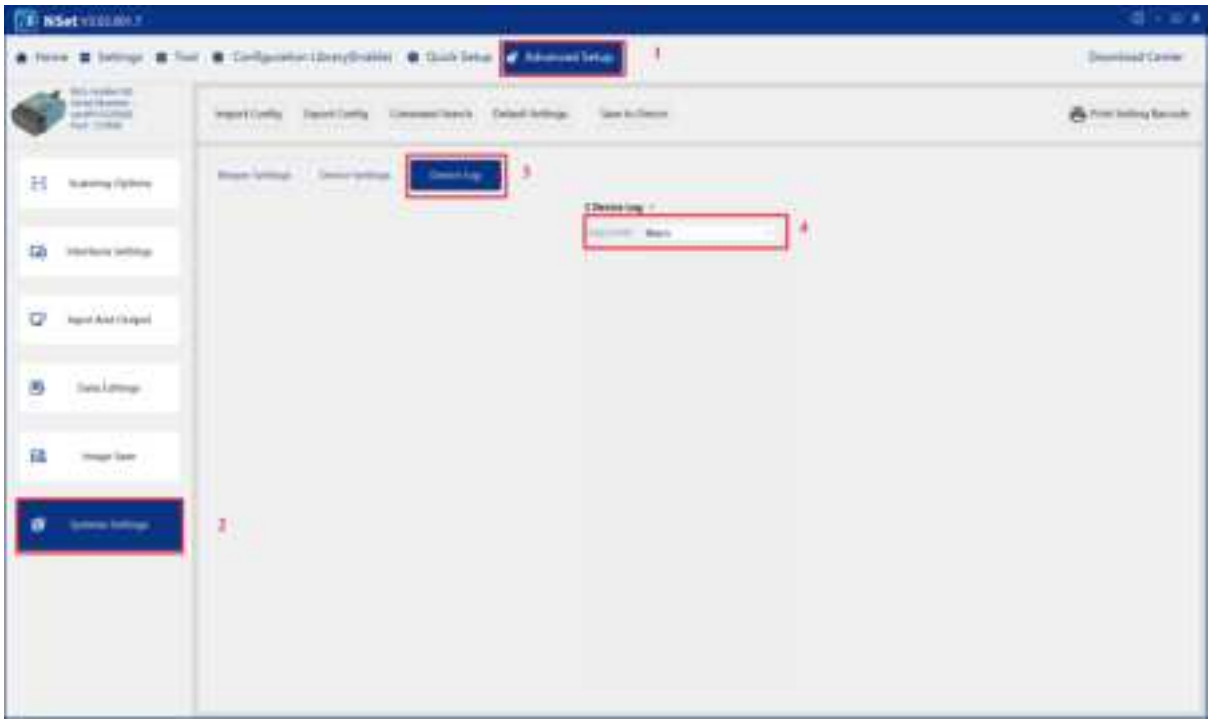
Parameter	Description	Setting Range	Default Value
Good Read LED	Enable or Disable the Good Read LED	0：Disable 1：Enable	1
Good Read LED Duration(ms)	Sets the amount of time the Good Read LED remains on	Short：20ms Medium：120ms Long：220ms	

		Custom	
Illumination Eye Protection Mode	Controls the lighting to eliminate flicker and provide a steady illumination.	5-15	
Aimer	Sets the status of the aimer.	Disable/Enable/Always on	Enable

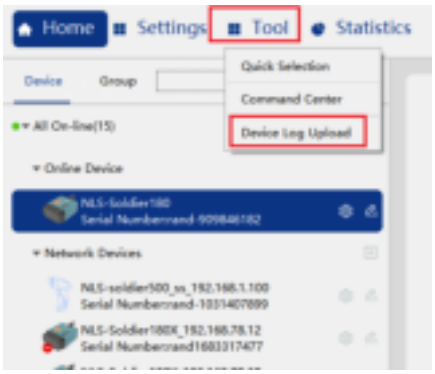
5.6.3 Device Log

The scanner supports two types of log: offline log and online log. All levels of online log can be viewed via Nset. Offline logs only store Assert, Error, and Warn level events in flash memory.

NSet can be used to view online logs (this function is only available when using network communication; it is not supported via other interfaces).



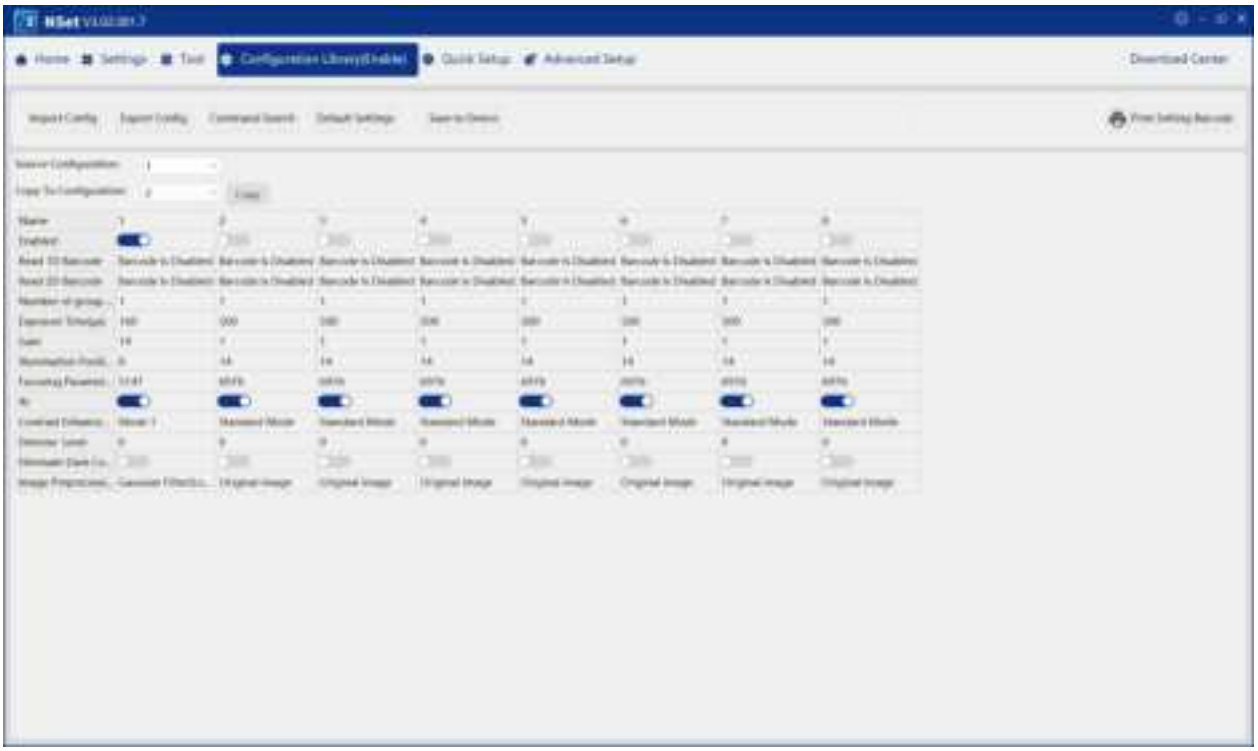
Logs are saved to the specified path.



Parameters	Function
Assert	Used to log program crashes
Error	Used to log failures and errors
Warn	Used to log temperature and peripheral monitoring data.
Info	Not visible to users, for internal development use only
Debug	Not visible to users, for internal development use only

6 Configuration Library

"Configuration Library" provides multiple sets of specific exposure, gain and focus parameters for different types of barcodes on the business line and complex scenarios that need to switch between different parameter combinations. The barcode scanner can poll the set parameter combinations until decoding is successful.



The configuration library parameters are shown in the image below:

Parameters	Function
Name	The library can be assigned any name, and the assigned name will remain unchanged even after reconfiguration.
Enabled	Enable or disable this Configuration Library, when using the Configuration Library to read barcodes, all enabled libraries can be polled in turn. Note: If no libraries are enabled, the scanner will not function properly for image capture and decoding.
Read 1D Barcode	Set the symbology for 1D barcode to be read
Read 2D Barcode	Set the symbology for 2D barcode to be read
Number of group retries	The number of re-reads allowed when the current Configuration Library is used to read the barcode unsuccessfully. If the barcode is not recognized after the number of times, it will poll to the next Configuration Library to read the barcode.
Exposure Time(μs)	Set the exposure time for image capture, in microseconds.
Gain	Set the gain value for image capture.
Illumination Position	Configure the position of the illumination LED
Focusing Parameters	Scanner focus parameter. It remains consistent across all configuration libraries.
AI	Enable or disable the AI function.
Contrast Enhancement	Set the contrast enhancement mode
Denoise Level	Set the image denoising strength (range: 0–7). Higher gain values require higher denoising levels.
Eliminate Dark Corner	Enables this function when decoding near the FOV boundaries.
Image Preprocessing	Set the type of image preprocessing

Note: If a learned library is copied to another library, all internal learning results in the algorithm library will be cleared. Reconfiguring the symbology will also reset all internal configurations of the algorithm library.

Chapter 3: Application Details

7 Application Examples

7.1 Device Debugging

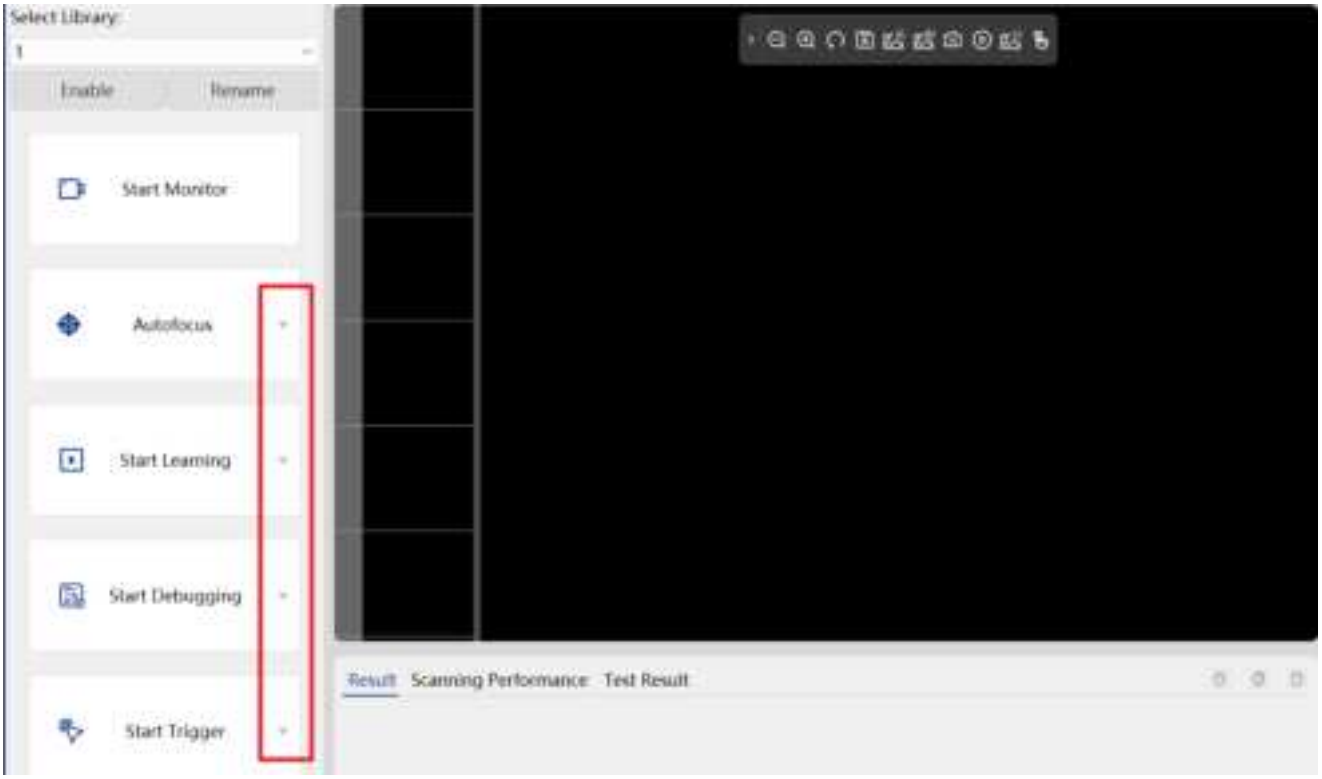
This section introduces the recommended steps for debugging the device effectively.

Step1: Auto Configuration

First, open the Monitor and place the barcode at the center of the field of view. Use the brightness slider on the right side of the screen to adjust the brightness to an appropriate level. Then, click "Autofocus"- > "Start Learning". Once learning is successful, the learned parameters will be saved to the selected configuration library. Finally, click "Start Debugging" to test the performance of the current configuration library.

If you need to save different focus parameters, please refer to the section: Auto Focus.

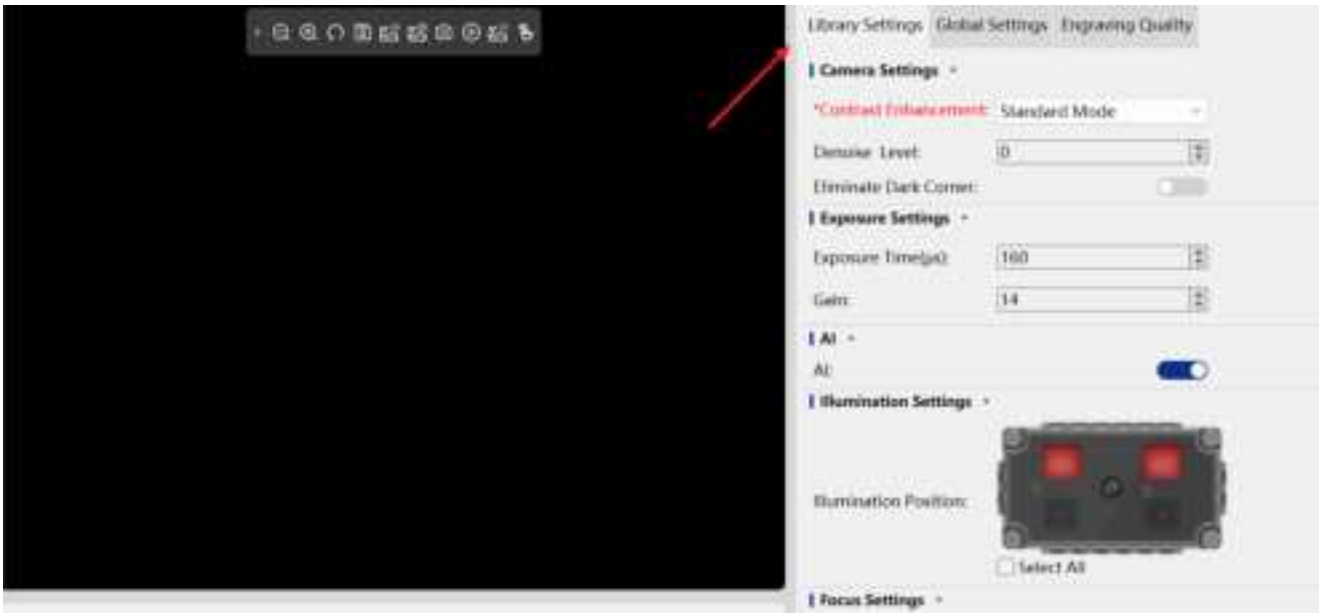
If auto learning fails or more advanced learning features are required (such as multi-barcode learning), refer to the section: Auto-learning.



Step2: Manual Optimization

If the auto-learned configuration parameters are unsatisfactory, you can manually adjust parameters such as exposure time and illumination position in the Library Settings. Manual optimization is saved in real-time, with no need to click a save button. You can also click Start Debugging simultaneously to view the effect of manual optimizations.

Note: Manual optimization only modifies the parameters in the currently selected configuration library. If you are satisfied with the parameters after Auto-learning, this step can be skipped.



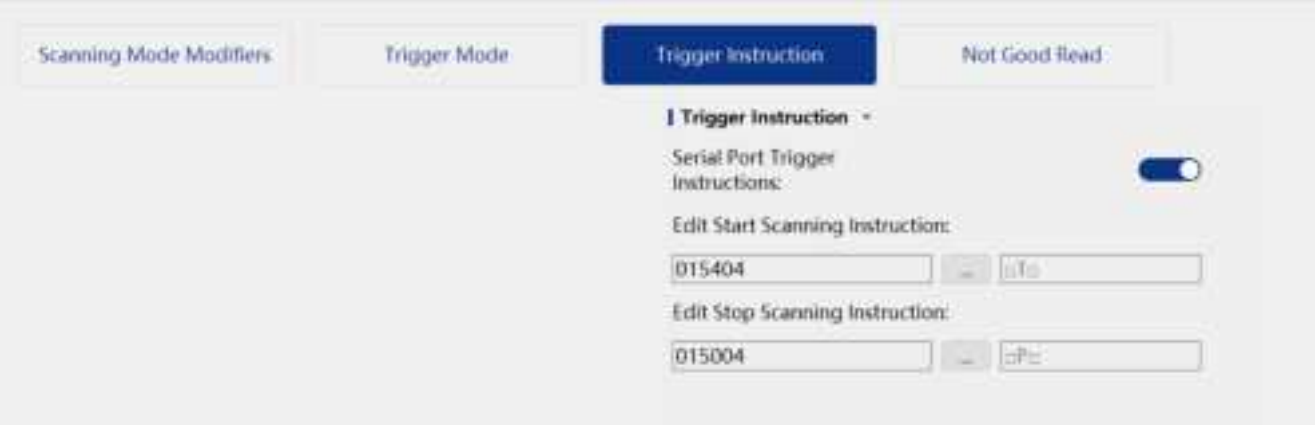
Step3: Scanning Settings

After completing Step 1 and Step 2, the scanner should be able to successfully read the barcode. Step 3 will explain how to activate the scanner and configure the scanning settings.

The scanner supports three scanning modes: Standard Mode, Continuous Mode, and Burst Mode. Select the appropriate mode in scanning mode modifiers based on your actual needs.

The scanner supports two trigger modes: Level Trigger and Pulse Trigger. Select the appropriate trigger mode to activate the scanner.

To trigger the scanner to read barcodes via command, enable and configure the relevant parameters in the trigger instruction settings.



Step4: Communication Settings

The scanner supports RS-232, USB, and Ethernet communication. Configure Communication settings as follows:

1. RS-232 Communication Settings

Enable the RS-232 output channel in the Output Channel Settings. Then, configure the RS-232 settings to match those of the PC (data receiving end). Make sure to select the correct COM port and ensure that all parameters are consistent during communication.

2. USB Communication Settings

To enable USB communication, activate the USB output channel in the Output Channel Settings. You can choose between USB HID Keyboard Output or USB CDC COM Port output.

When the USB port is available on the scanner's interface cable, USB settings will take effect automatically.

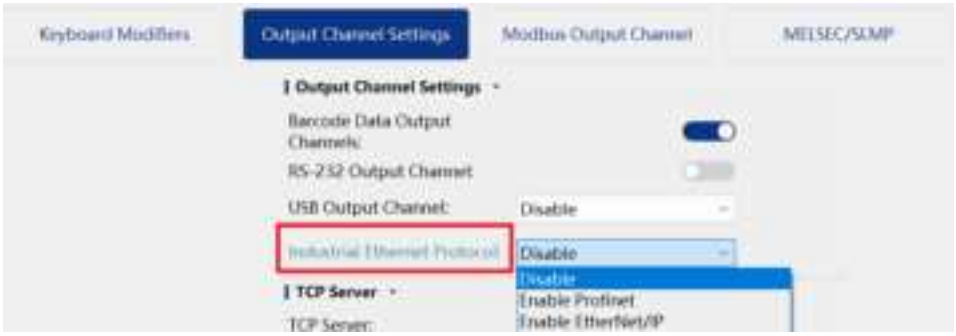
If the interface cable only provides an RS-232 or Ethernet port, you will need an RS-232-to-USB or Ethernet-to-USB adapter. In such cases, USB Keyboard output must be used in combination with the NLKbw tool for proper functionality.

3. Ethernet Communication Settings

Ethernet communication supports TCP/IP, UDP, FTP, Modbus TCP, Ethernet/IP, Profinet, and MELSEC\SLMP. Note that the "client" and "server" mentioned in the Nset refer to the scanner itself.

You can set TCP/IP, UDP, Modbus, and MELSEC\SLMP as shown below:

Enable the Ethernet/IP and Profinet as shown below:



For communication between the PLC and the scanner, please refer to the *Industrial Barcode Scanner Communication Protocol Handbook*. This document can be obtained from local sales or technical support.

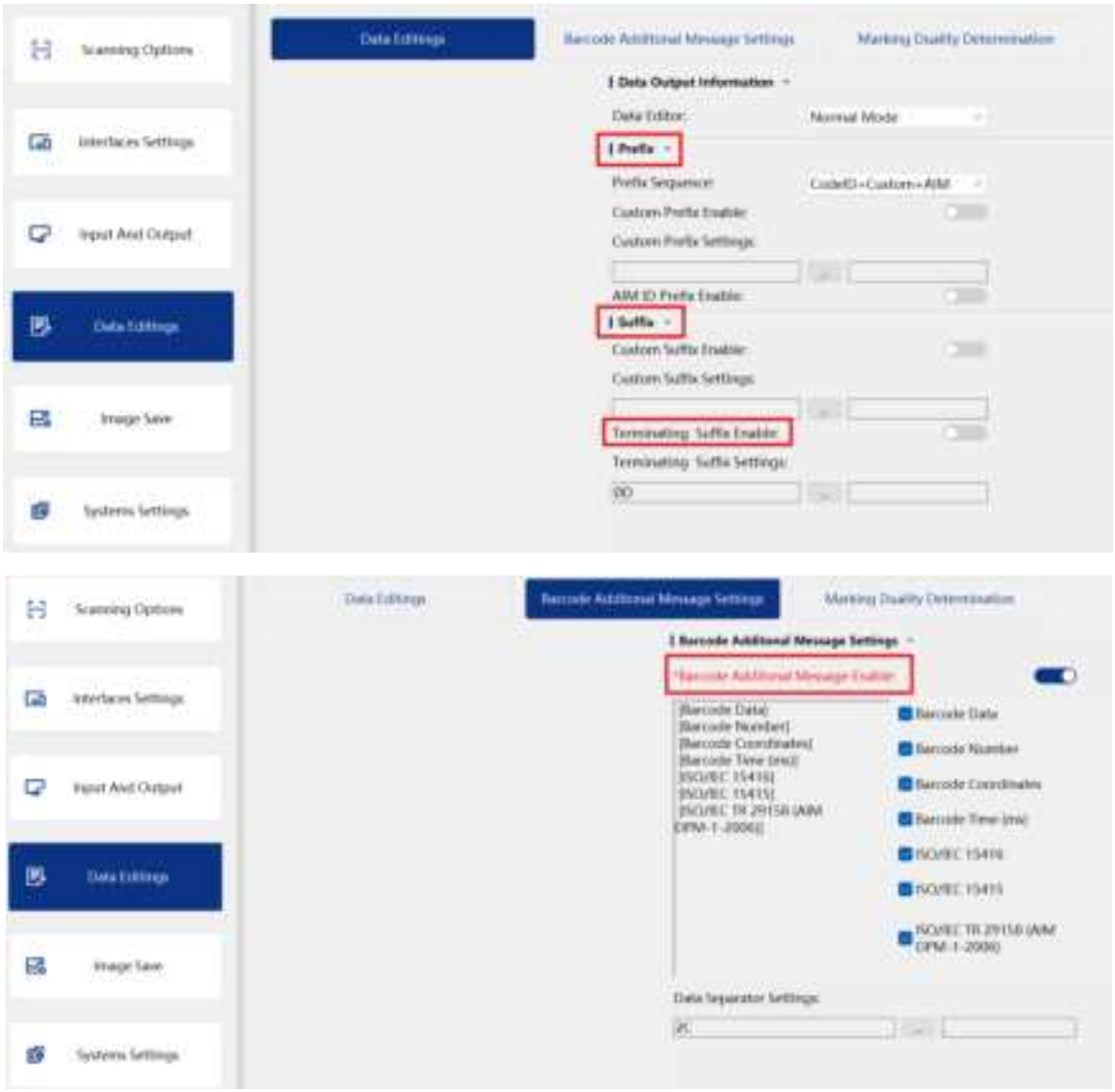
Step5: Input & Output

After completing Step 1 to Step 4, the scanner is able to successfully read the data and transmit it to the data receiver. This chapter introduces how to define and use the scanner's I/O ports.

Tap "Advanced setup" - > "input and output", then setting the parameters for input and output based on actual application needs.

Step6: Data Editings

After decoding the barcode, the scanner can format the output data as needed. Prefixes, suffixes, terminators, and Barcode Additional Message can be configured directly in Nset, as shown below.



To perform more advanced data formatting, you will need to enable script mode.

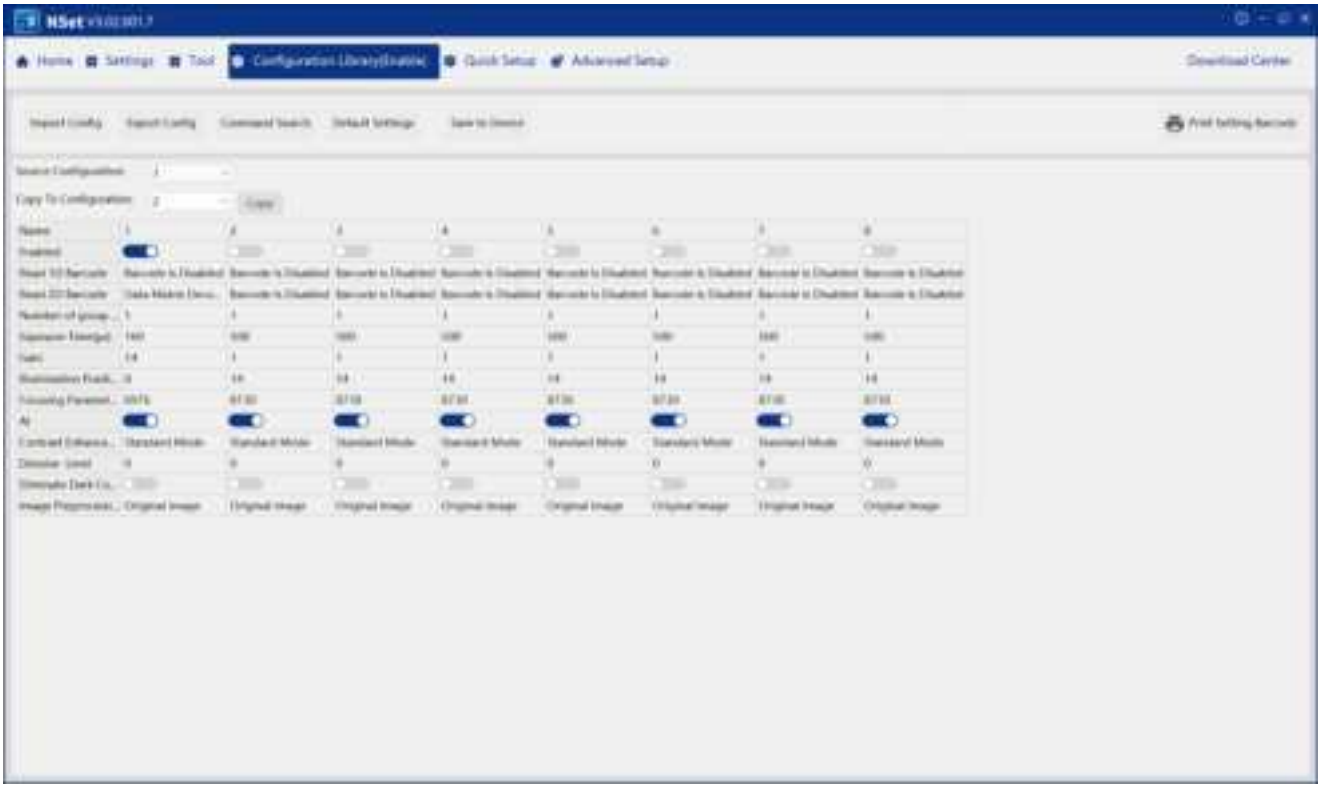
7.2 Auto-learning

The Auto-learning feature allows users to configure barcode image parameters with a single click on the icon

in the Nset app, or by long-pressing the device button. The setup is simple and user-friendly.

Auto-learning Scope

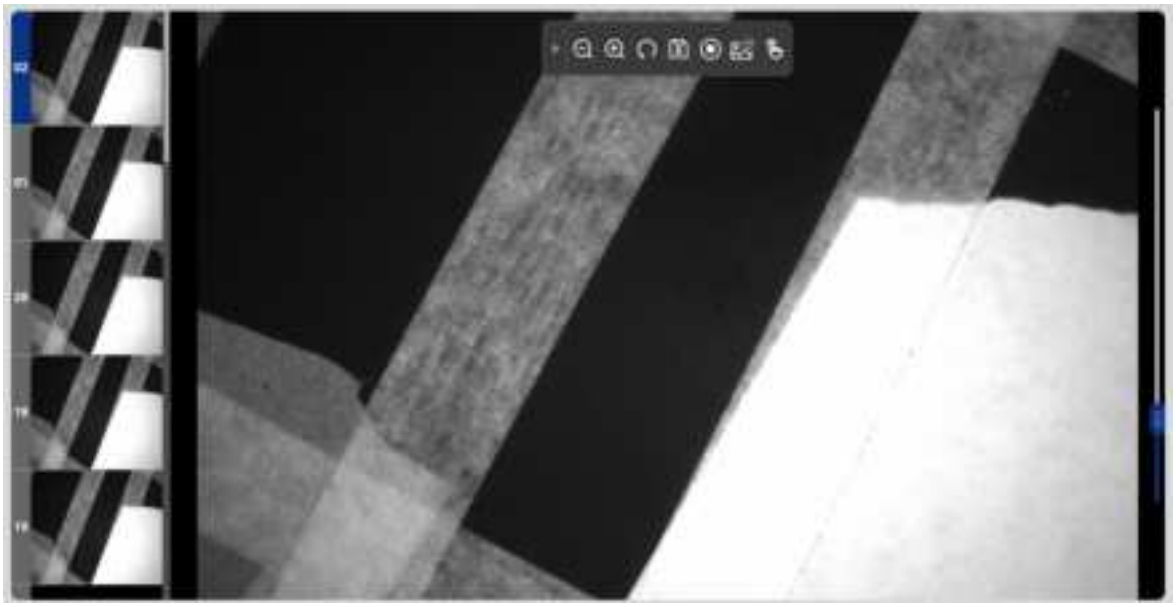
The Auto-learning feature covers the following parameters: Symbology, Exposure Time, Gain, Illumination Position, AI Enable/Disable, Contrast Enhancement, Denoise Level, Eliminate Dark Corner, and Image Preprocessing.
The learning results will be applied to the selected enabled configuration library. Parameters not included in the Auto-learning scope must be adjusted manually.



Auto-learning Failure

If the Auto-learning fails, you can try the following actions:

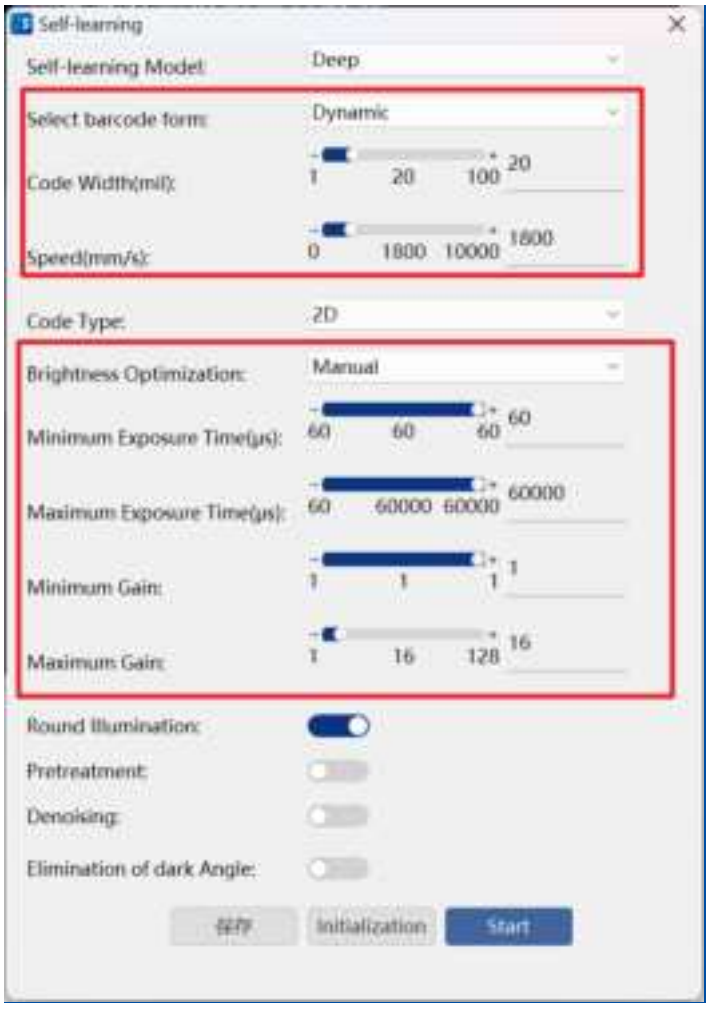
1. Ensure that the installation is correct.
2. Ensure the scanner is properly focused.
3. Check if the barcode is clearly visible on the monitor, and adjust the brightness using the brightness slider.
4. Go to "Start Learning" -> "Brightness Optimization", and ensure that the exposure time and gain settings there are not restricting the learning process.



Configure the Auto-learning Parameters

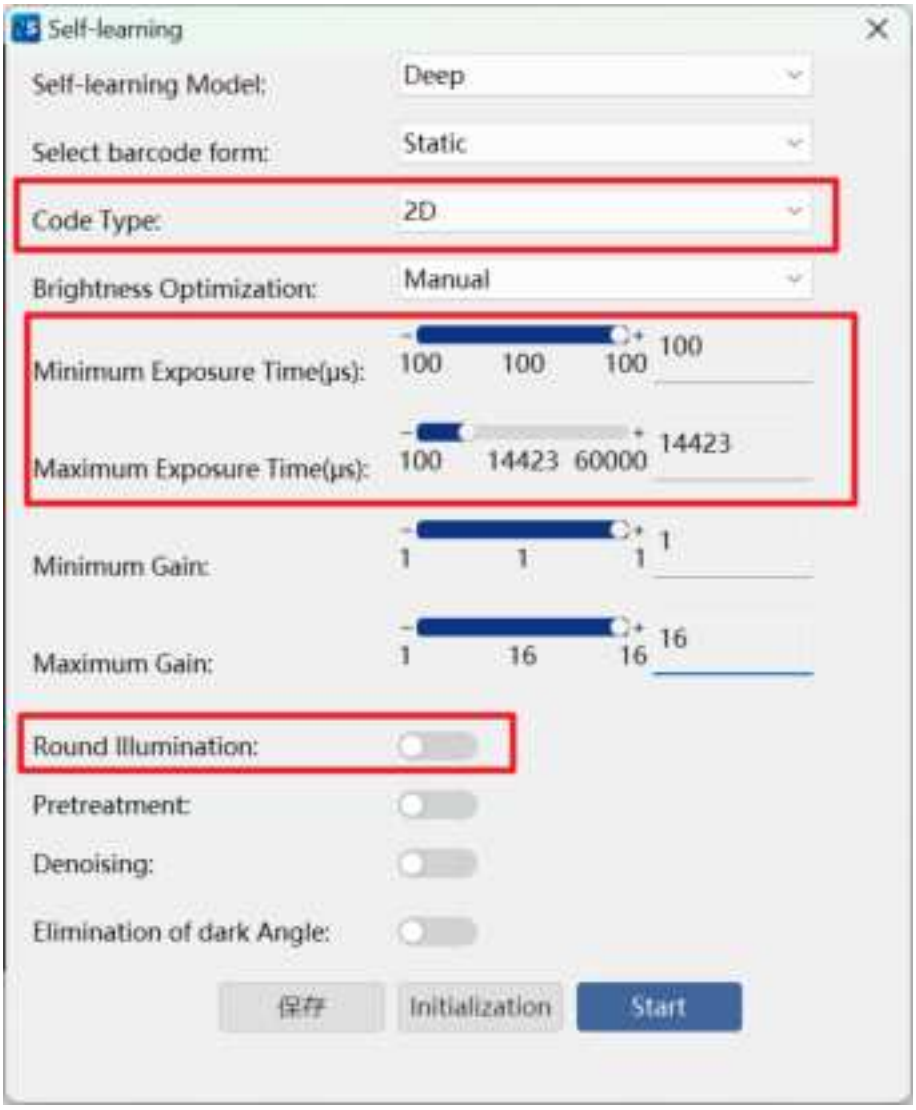
When the barcode form is set to "Dynamic", it does not mean that the scanner performs Auto-learning while the object is moving. To ensure proper Auto-learning, the object must remain stationary during the process.

Dynamic Auto-learning is designed to help the scanner achieve successful barcode reading at a specified moving speed. By entering the code width and the desired speed, the scanner will automatically adjust exposure time and gain to meet the requirements. Alternatively, users can manually define brightness to optimize performance.



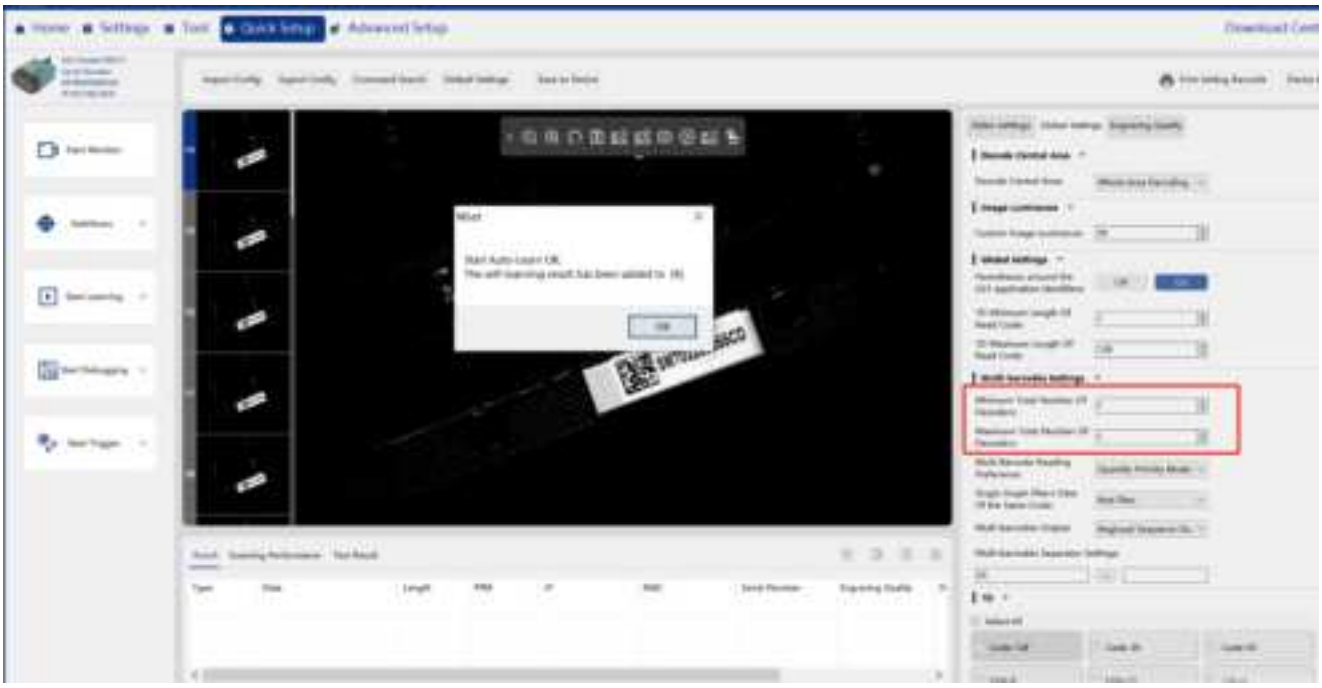
Optimizing Auto-learning Speed and Accuracy

In the Auto-learning function, defining specific conditions can significantly improve both the learning speed and accuracy



Multi-barcode Auto-learning

The Auto-learning function supports simultaneous learning of multiple barcodes. The Multi-barcode Auto-learning process is considered successful once minimum total number of decoders is reached.



When using different configuration libraries to decode barcodes, with each library decoding multiple barcodes, first set the number of decoders according to the requirements of each library, and then sum the numbers during the triggering session. If the total number is set directly, auto-learning will fail. See the examples below.

Configuration Library 1 decodes 2 barcodes, and Configuration Library 2 decodes 3 barcodes. The total number of barcodes decoded during triggering is 5(assuming all 5 barcodes can be decoded).

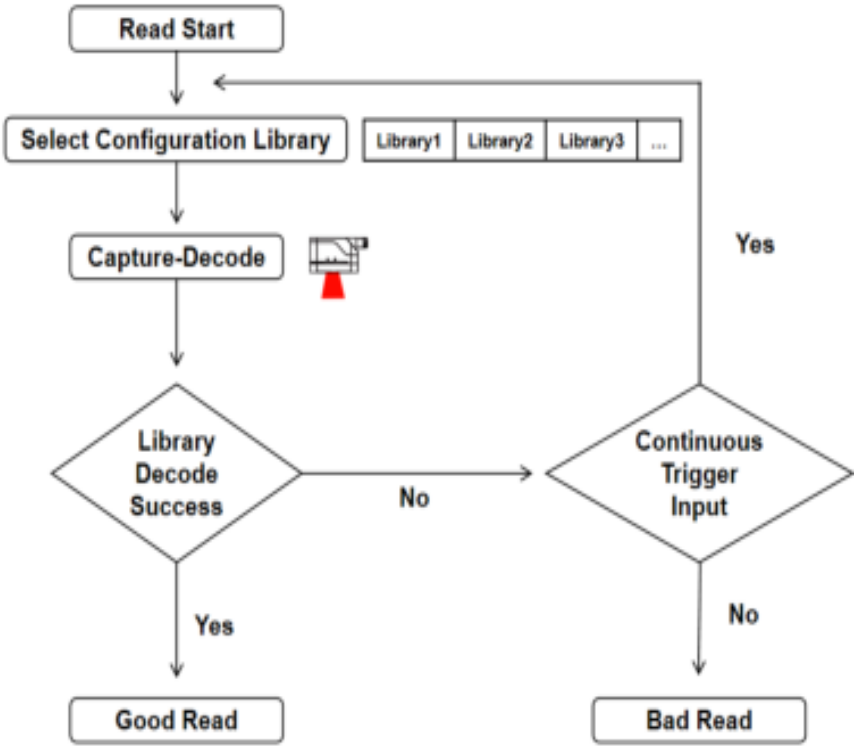
Configuration	Auto-learning Result of Configuration Library 1	Auto-learning Result of Configuration Library 2	Trigger Result
Set the total number of decoders for Configuration Library 1: Minimum value=2, Maximum value=2 Set the total number of decoders for Configuration Library 2: Minimum value=3, Maximum value=3 Set the total number of decoders during trigger: Minimum value=5, Maximum value=5	Auto-learning successfully decoded 2 barcodes	Auto-learning successfully decoded 3 barcodes	5 barcodes decoded successfully
Set the total number of decoders for Configuration Library 1: Minimum value=5, Maximum value=5 Set the total number of decoders for Configuration Library 2:	Auto-learning failed	Auto-learning successfully decoded 3	3 barcodes decoded successfully

Minimum value=3, Maximum value=5 Set the total number of decoders during trigger: Minimum value=1, Maximum value=5		barcodes	

7.3 Alternate Configuration Library

When a single set of parameters cannot meet customer requirements, the alternate configuration library feature can be used. This feature allows the scanner to switch between multiple configuration libraries during scanning. It is primarily used to adjust hardware parameters, such as illumination type, exposure time, and more.
For details, refer to Chapter 6 – Configuration Library.

Function Introduction



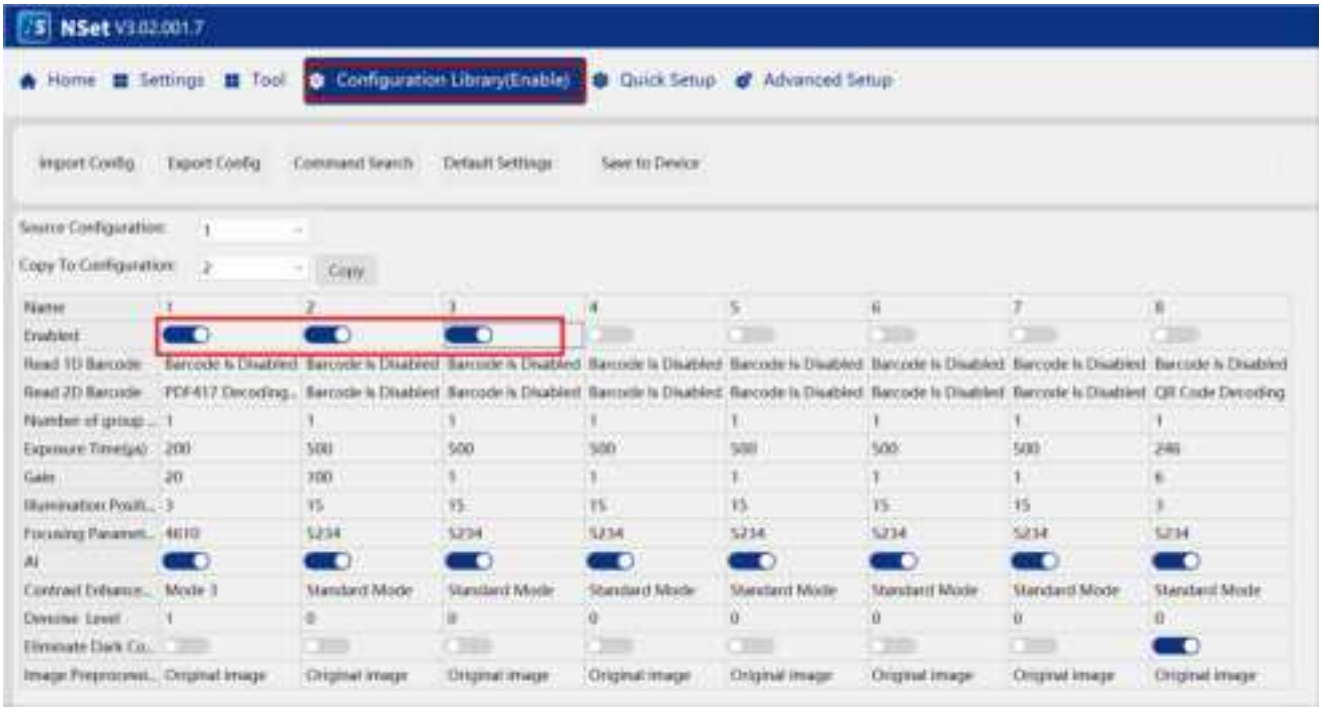
Each time the scanner is triggered, it follows the decoding sequence as outlined below:

1. If the previous trigger did not successfully decode the barcode, the scanner will move on to the next configuration library in the sequence.
2. If the previous trigger successfully decoded the barcode, the scanner will prioritize the configuration library that was used last time.

- A single configuration library is considered successful only if it can complete all tasks. For example, for multi-barcode (1-N, N≥2) reading, the configuration library must decode all N barcodes successfully.
- After changing the configuration library settings or performing a reset or restart operation, the scanner will re-execute the configuration libraries in sequence, starting from the configuration library with the lowest number.

Enable the Alternate Configuration Library

1. Go to the configuration library interface
2. Enable the configuration libraries for alternation. The scanner will automatically cycle through all enabled configuration libraries in ascending numerical order (Library 1 → 2 → 3...) during each scanning attempt.



Example

Scenario	Problem Analysis	Recommended Configuration
A customer sample includes a printed paper barcode and a low-contrast etched barcode that must be decoded in a single trigger.	A single exposure setting cannot reliably read both barcodes due to the large difference in contrast.	Learn each barcode into a separate configuration library and enable the Quantity Priority mode.

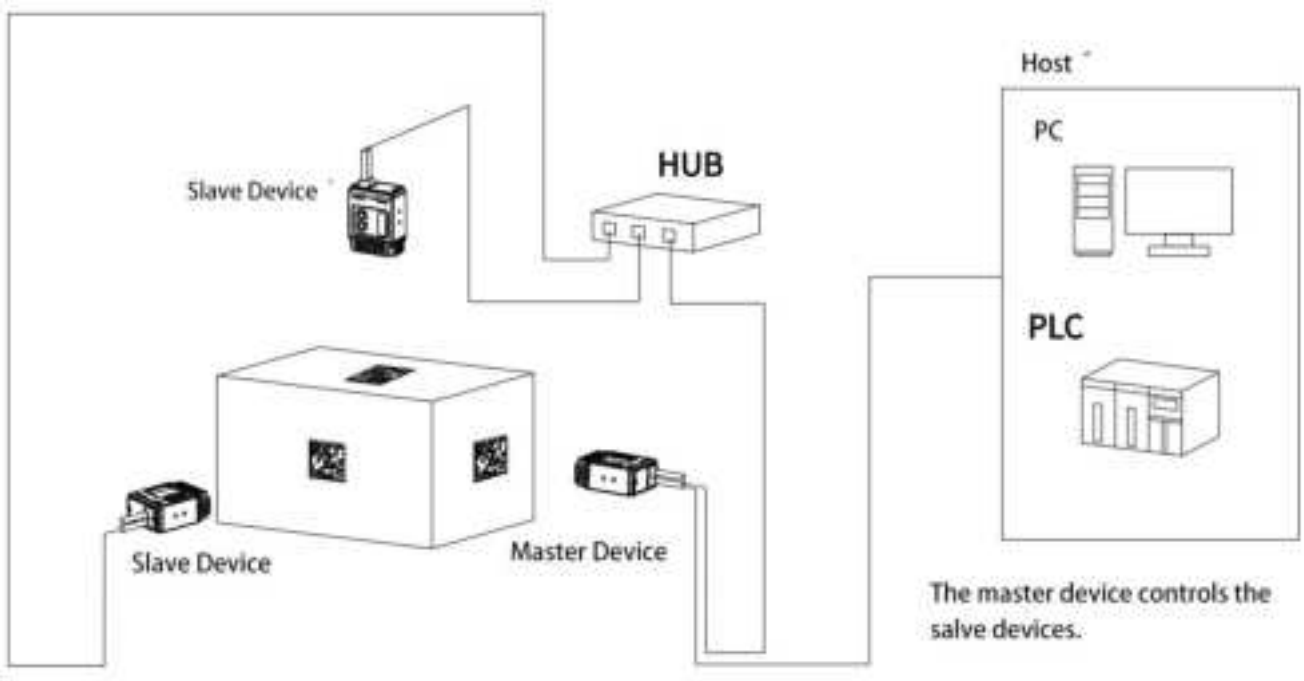
7.4 Device Group Configuration

Synchronous mode

This mode is designed for the following two scenarios:

1. The barcode position on the product is uncertain, requiring multiple devices to cover all possible surfaces.
2. A single device cannot cover the full required field of view, so multiple devices must be networked to expand the scanning area.

The Soldier180 supports networking of up to 8 devices. All devices can be triggered simultaneously for synchronized scanning. Once the required number of barcodes has been successfully decoded, the results will be consolidated and sent to the host.



Parameter	Function
Networking Barcode Priority Mode	After the slave device decodes, it will be sent to the host immediately. After the host summarizes all the barcodes, as long as the number of networking barcodes meets the "networking barcode maximum value", or reaches the "networking timeout" and the number of networking barcodes is greater than or equal to the "networking barcode minimum value", the barcode will be output, otherwise NG will be output.

Function Description

Function	Synchronous Mode
Trigger & Completion	1. Triggering the master device will simultaneously trigger the slave devices. 2. Triggering a slave device independently is supported, but its data will not be uploaded to the host.
Configuration Library & Scanning Mode	1. Master and slave devices use their own configuration libraries for image capture and decoding. 2. Master and slave devices operates under its own scanning mode.
Input & Output	1. When the master device is triggered via the input signal, the connected slave devices are triggered simultaneously. Slave devices can receive external triggers, but its data will not be uploaded to the host. 2. If both the master and slave devices are connected to output terminals, the master device will output data based on whether the number of networking barcodes meets the configured value. Each slave device will output data based on whether the number of its decoded barcodes meets the configured value.
Data Editing	1. Additional information is edited and applied individually on each device. 2. Script editing is applied individually on both master and slave devices; however, the master device will further edit the data received from the slave devices. 3. The output sequence of codewords is determined by the master device configuration.
OK/NG Judgment	The master device determines the OK/NG status based on whether the number of networking barcodes meets the configured value. If the configured value is met, it returns OK; otherwise, output NG.

Example of Reading Logic in Synchronous Mode

The master device decodes 3 barcodes, and the slave device decodes 2 barcodes. The networking mode is Synchronous Mode.

Number of Networking Barcodes	Number of Barcodes Configured in Master Device	Number of Barcodes Configured in Slave Device	Operating Status of Master Device	Operating Status of Slave Device
Minimum value =5 Maximum value =5	Minimum value =3 Maximum value =3	Minimum value =2 Maximum value =2	Exit after decoding 3 barcodes	Exit after decoding 2 barcodes
Minimum value =2 Maximum value =2	Minimum value =3 Maximum value =3	Minimum value =2 Maximum value =2	Output the first 2 decoded barcodes, which can be from any device	

Minimum value =5 Maximum value =5	Minimum value =3 Maximum value =3	Minimum value =2 Maximum value =3	Exit once the configured number of networking barcodes is reached; it does not wait for the trigger release or the completion of slave device tasks.	
Minimum value =6 Maximum value =6	Minimum value =3 Maximum value =3	Minimum value =2 Maximum value =2	Continues decoding until trigger action ends	Exit after decoding 2 barcodes
Minimum value =6 Maximum value =6	Minimum value =3 Maximum value =3	Minimum value =3 Maximum value =3	Continues decoding until trigger action ends	Continues decoding until trigger action ends

Note: The master device determines task completion based on the number of networking barcodes set, while the slave device determines completion based on its own barcode number settings. The slave device will only exit the reading operation once it has completed its task. Otherwise, the master device will decide when the slave device exits, based on the completion of the networking barcode task.

Operating Procedure

Scenario Example: The customer requires two devices to operate in a group. The master device decodes 3 barcodes, and the slave device decodes 2 barcodes, total output 5 barcodes.

Step 1:
Connect the sensor to the scanner designated as the master device. Connect both the master and slave devices to a switch using Ethernet cables. Then connect the switch to the PC (data receiving terminal) with another Ethernet cable.

Step 2:
Debug the master scanner. Set both the minimum and maximum barcode number to 3. When triggered, the History Results section will display 3 decoded barcodes.

Step 3:
Debug the slave scanner. Set both the minimum and maximum barcode number to 2. When triggered, the History Results section will display 2 decoded barcodes.

Step 4:
Group the two devices in NSet. Set the both the minimum and maximum networking barcode number to 5.

Step 5:
Return to the master device's debugging interface and ensure that grouping is successful. Triggering the master device, the Results section will display 5 decoded barcodes.

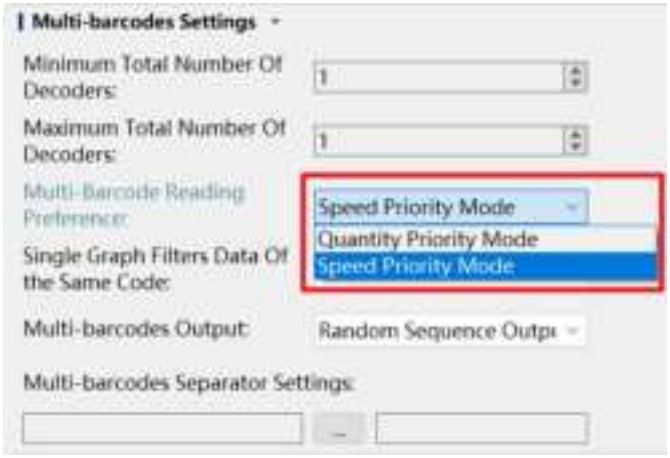
Note: Each scanner should be configured and debugged individually with the required number of barcodes before grouping. Grouping the devices before individual configuration is completed may cause discrepancies in the number of barcodes decoded, leading to decoding failure on the master device.

7.5 Multi-barcode Reading Preference

In scenarios where the scanning field of view is insufficient to cover all barcodes, or a single library configuration cannot decode all the barcodes. Suggest to use the speed priority mode in the multi-barcode reading preference setting. This feature enables the scanner to decoding multiple barcodes by combining the results from multiple images

Settings

In the multi-barcode settings, setting the multi-barcode reading preference to "Quantity Priority Mode" will combine decoding results from multiple images. This feature also applies when using the alternate configuration library function.



Feature Introduction

- **Quantity Priority**

1. During the trigger, if the number of decoders reaches the maximum output limit, the reading process stops in non-continuous modes (Standard, Burst).
2. If the trigger ends and the number of decoders meets the minimum output limit, the data is output; otherwise, "NG" is output.

- **Speed Priority**

1. During the trigger, if the number of decoders meets the minimum output limit, the reading process stops in non-continuous modes (Standard, Burst).
2. If the trigger ends and the number of decoders meets the minimum output limit, the data is output; otherwise, "NG" is output. In continuous mode, if the minimum output limit is never met throughout the entire trigger duration, "NG" is output.

Example

Scenario 1: A product has 5 barcodes, but only 1 of them is required by the customer. The goal is to read as many barcodes as possible while ensuring a smooth reading experience.
Configuration: Speed Priority, Minimum Total Number of Decoders = 1, Maximum Total Number of Decoders = 5

Scenario 2: The number of barcodes on the product varies; some have 2 barcodes, while others have 3. All barcodes present on the item need to be decoded.
Configuration: Quantity Priority, Minimum Total Number of Decoders = 2, Maximum Total Number of Decoders = 3

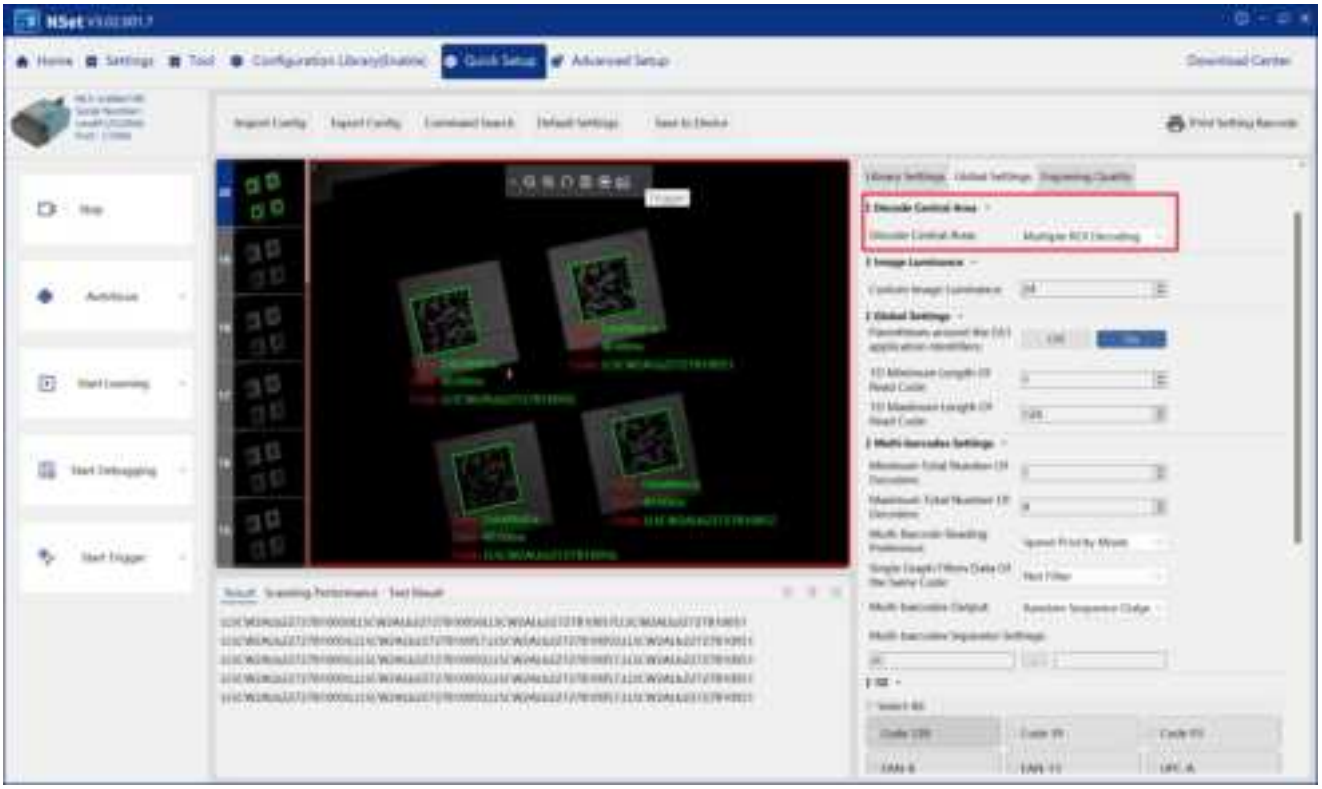
Scenario 3: A product has 5 barcodes, which require decoding across two configuration libraries or multiple images. All barcodes must be decoded.
Configuration: Quantity Priority, Minimum Total Number of Decoders = 5, Maximum Total Number of Decoders = 5

Scenario 4: A product has 5 barcodes and require all barcodes decoded successfully using a single configuration library.
Configuration: Speed Priority, Minimum Total Number of Decoders = 5, Maximum Total Number of Decoders = 5

7.6 Multiple Area Decoding(ROI)

The Multiple Area Decoding (ROI) function can be used in complex barcode environments. Such as when multiple barcodes are within the same field of view, but only one or some barcodes need to be decoded. Each frame selection area must completely cover the range that needs to be decoded. With the multi-code setup, the multi-code can be accurately read.

Select "Multiple Area Decoding", the mouse will turn into a cross cursor, then you can directly select the specified rang in the imaging area.



Add multiple regions

Right-click in the imaging area and select "adding multiple regions".

Enter the number of horizontal/vertical regions to quickly add multiple reading areas

Start Sorting Regions Numbers

After clicking "Start Sorting Regions Numbers", click the areas in the specified order, that is, the areas are sorted according to the click order. For example: Click each area in the order of the original number "7→5→8→9→6→4→1→2→3", that is, the sorting changes are as follows:

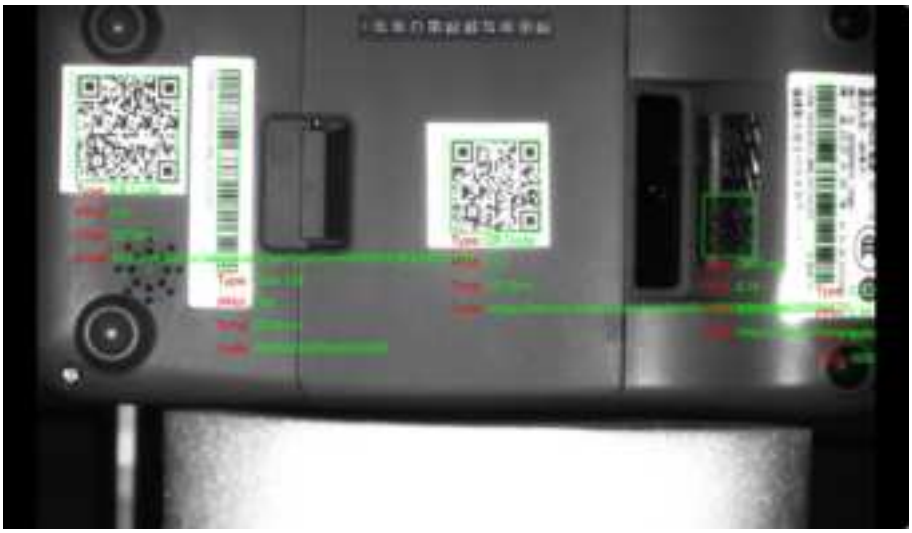
Right-click again, and click "Stop sorting Regions Numbers" to exit sorting.

7.7 Multi-barcodes Output

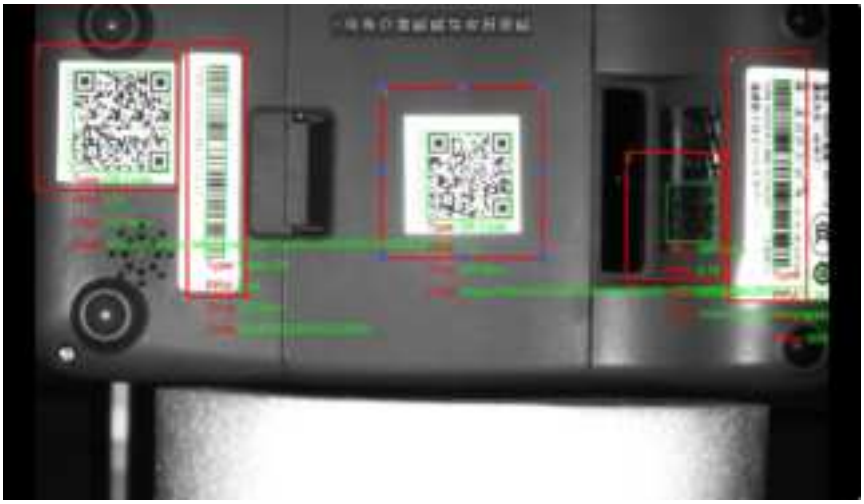
1 Double-click on the device to be configured, and go to "Quick Setup"

2 Click on the "Global Parameters" to access the settings page.

3 Scroll down to select Multi-Barcode Output mode, and configure parameters for vertical direction, horizontal direction, and barcode type based on actual needs.



The first mode is Random Sequence Output



The second mode is Regional Sequence Output

The third mode is Fixed Sequence Output.

Higher-priority settings take precedence over lower-priority ones.

Fixed Sequence Output > Regional Sequence Output > Random Sequence Output

7.8 Transmit Not Good Read Message in Regional Sequence Output

Configuration Requirements:

Select "Multiple ROI Decoding" in Decode Central Area.

Select "the Regional Sequence Output" in Multi-Barcode Output Mode.

Enable the "Transmit Not Good Read Message" with default "NG" display.

When the number of ROIs is equal to the maximum number of decoders, and some ROIs fail to decode successfully, the output data will be sorted in ascending order by ROI index, with "NG" filled in the positions corresponding to failed ROIs.
Example:

Total Number of Decoders	Number of ROIs	Decoded Barcodes per ROI	Failed ROI Position	Output Format (Ascending Order)
Minimum Value =1 Maximum Value =4	4	1	ROI3	Barcode, Barcode，NG, Barcode

If the number of ROIs does not match the Total Number of Decoders limit, the data output format is as follows:

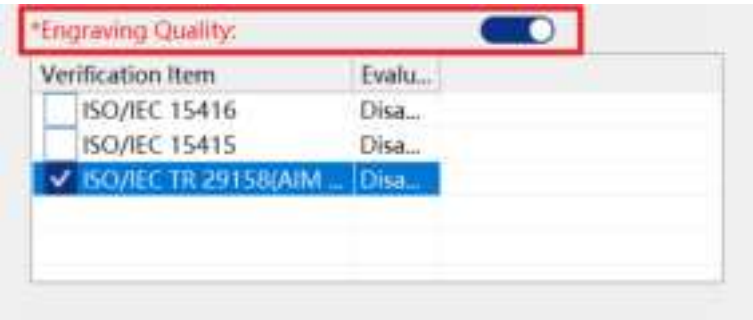
Total Number of Decoders	Number of ROIs	Decoded Barcodes per ROI	Output Format (Ascending Order)
Minimum Value =3 Maximum Value =4	5	ROI1 decodes 5, other ROIs decode 0	4 Barcodes + 4 NG
	5	ROI2 decodes 4, other ROIs decode 0	NG + 4 Barcodes + 3 NG
	5	ROI5 decodes 3, other ROIs decode 0	4 NG + 3 Barcodes
	5	ROI1 decodes 2, other ROIs decode 0	2 Barcodes + 4 NG
	2	ROI1 decodes 2, other ROIs decode 0	2 Barcodes + 1 NG

7.9 Engraving Quality Grade

Engraving Quality is used to verify the quality of barcodes scanned based on the engraving quality standards specified by international organizations. The supported standards are as follows:

Standard	Application Range
ISO/IEC 15416	For 1D barcodes
ISO/IEC 15415	For 2D barcodes printed on labels
ISO/IEC TR 29158(AIM DPM-1-12006)	For 2D DPM barcodes

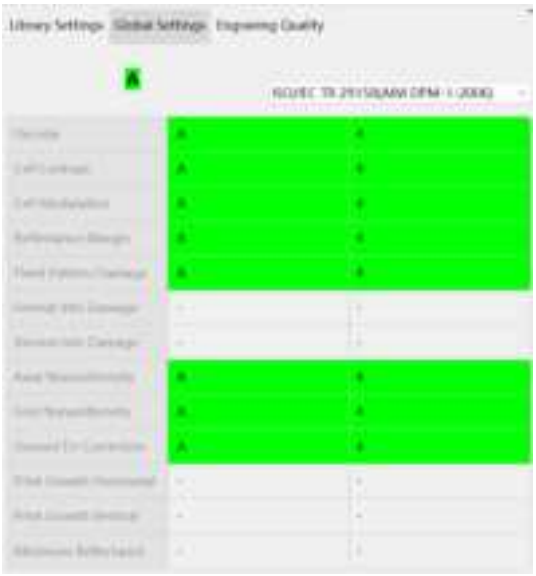
Enable/disable the engraving quality:



Once engraving quality is enabled, the engraving quality grade can be viewed in the Results section after a successful decode.

Type	Data	Length	PPM	Engraving Quality
DataMatrix	FW6R231AA0JX	12	4.94	B
DataMatrix	FW6R231AA0JX	12	4.94	B
DataMatrix	FW6R231AA0JX	12	4.94	B
DataMatrix	FW6R231AA0JX	12	4.81	B
DataMatrix	FW6R231AA0JX	12	4.94	B
DataMatrix	FW6R231AA0JX	12	4.94	A

Go to "Quick Setup" - > "Engraving Quality" to view detailed engraving quality grade. This feature applies to single barcodes only.

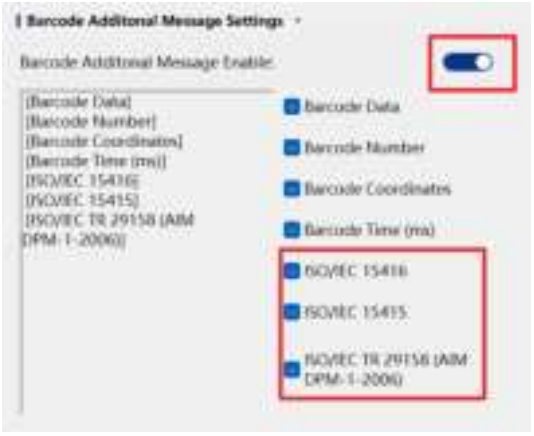


With the Engraving Quality feature enabled, the following two functions can be implemented:

1 Barcode Additional Message

2 Marking Verification Success Indicator

Barcode Additional Message



Marking Verification Success Indicator

Verification Item	Evalu...
☐ ISO/IEC 15416	Disa...
☐ ISO/IEC 15415	Disa...
☑ ISO/IEC TR 29158(AIM ...	Disa...

7.10 Enhancing Motion Tolerance

To improve a scanner's motion tolerance, consider optimizing the following parameters:

Trigger Timing

Ensure that the barcode is close to the center of FOV when triggered by the sensor.

Decoding Area

The scanner should be configured to decode barcodes throughout the entire field of view, rather than relying on a fixed position. Expanding the decoding area enhances the decoding success rate and can typically be achieved by adjusting the sensor position or adding configuration libraries.

Exposure Time Adjustment

Proper exposure time is critical for motion tolerance. If the exposure time is too short, the image will be dark and difficult to decode. If it is too long, the captured barcode image may appear smeared or blurred. The appropriate exposure time can be estimated using the following formula:

t = (25.4 × barcode width in mils) / v

t: exposure time in microseconds (μs)

v: scanning speed in meters per second (m/s)

To validate the calculated exposure time, use the Auto Learning feature (see Section 7.1). Set the barcode type to Dynamic, input the target exposure time, and start the learning process. If learning succeeds, the exposure time is considered valid.

Multi-Region Decoding (ROI)

The Multi-Region Decoding function allows you to focus on target decoding areas while excluding irrelevant zones, thereby reducing overall decoding time.

Appropriate Trigger Mode

Use pulse mode if the sensor's level signal is shorter than 20 ms.

7.11 Update Firmware

Before upgrading the firmware version, please download the firmware to the local.

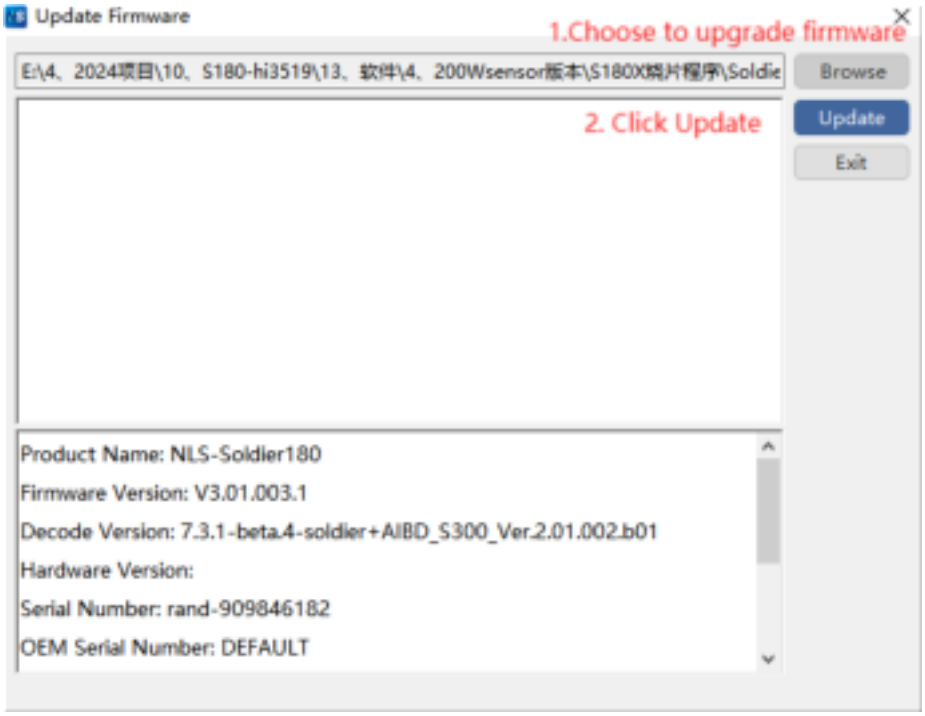
USB CDC recommended

1 Click "Upgrade Firmware" to enter the Upgrade Firmware configuration window;

2 Click "Browse" to select the firmware file to be upgraded;

3 Click "Update" to update according to the selected firmware file.

4 During the upgrade process, please ensure that the device is online in real time.



Note: During the firmware upgrade process, wait for about 1 minute to allow the device to restart. The restart is considered successful when the beeper sounds and the corresponding product model is refreshed in the Nset. Additionally, you can check the firmware version on the Nset homepage to ensure its correctness.


Update Firmware
Browse
Update
Exit
Update Appl Block
Send Update Command
Set Param
Send Data
Burning in Firmware
Update OK
Note: Do not remove the device until the device is successfully restarted.
OEM Serial Number: DEFAULT
Kernel Version: V2.00.003.4
LightBoard Version: V1.00.002
Port:COM72


Product Name: NLS-Soldier180
Firmware Version: V3.01.002.1
Decode Version: 7.3.1-beta.4-soldier+AIBD_S300_Ver.2.01.002.b01
Hardware Version:
Serial Number: rand-909846182
OEM Serial Number: DEFAULT
Kernel Version: V2.00.003.4
LightBoard Version: V1.00.002
Port: COM72

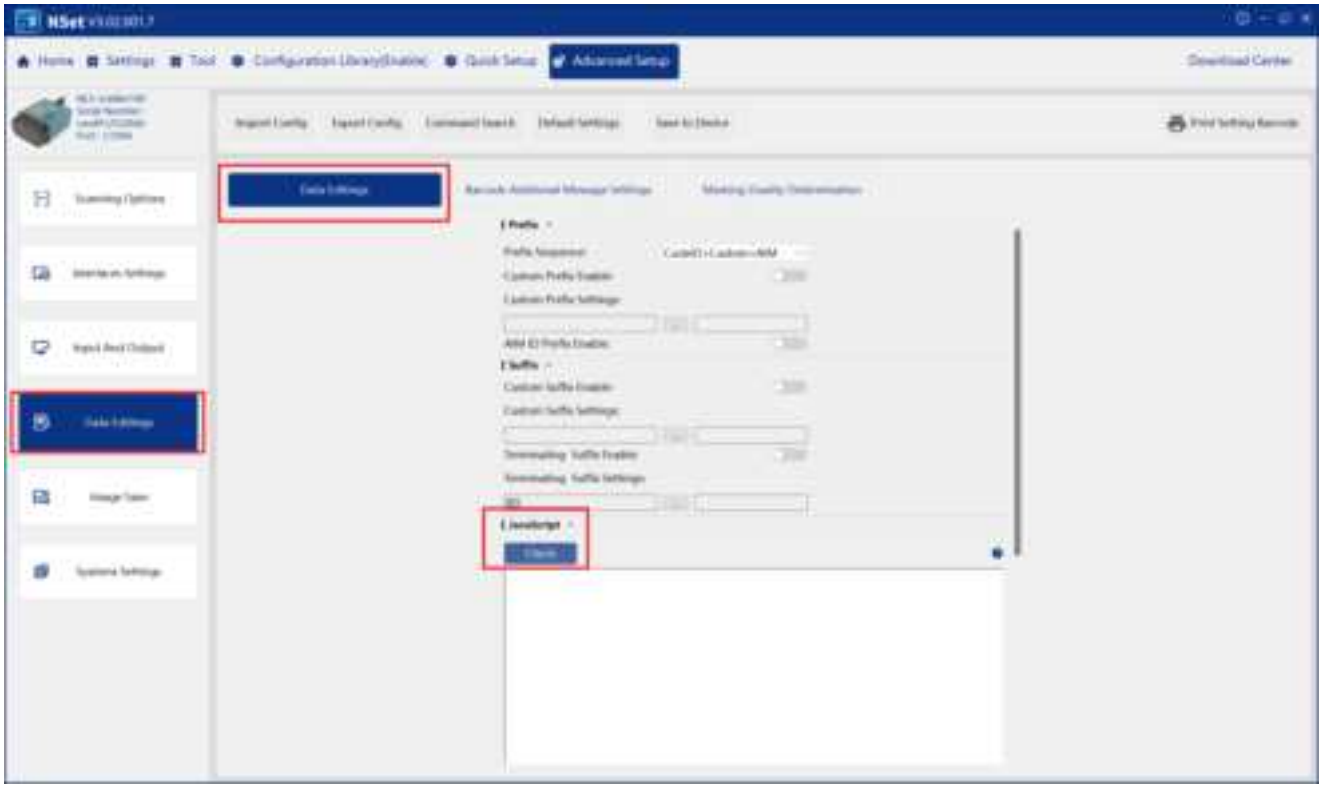
7.12 Script Editing

The script editing function allows you to customize and output the decoded data in any desired format.

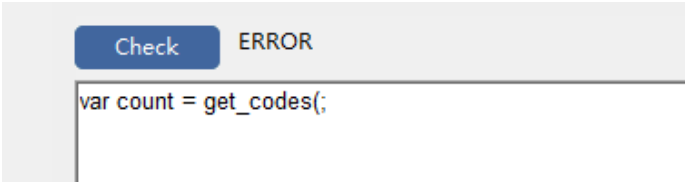
Double-click the Soldier180 icon and navigate to "Advanced Setup "- > "Data Editing", then set "Data Output Information" to "Advanced Mode", then you can to entering the JavaScript script

Click "" to get the script help document.

After writing the script, click "Check" to perform a basic syntax check in Nset.



Check whether the syntax is correct, as shown below:



For information on how to write javascripts for industrial products, please refer to the "API Instruction" section below.

API Instruction：

Method	int get_codes()
Function Description	Retrieves the total number of current code words, suitable for single and multiple codes.
Paramaters	None
Return value	Integer, returns the number of code words

Method	string get_decMsg(int n)
Function Description	Retrieves the nth code word from all code words
Paramaters	n: Integer, range 0-N; the nth code word from all code words
Return value	String; the nth code word

Method	void get_codeType(int n)
Function Description	Retrieves the code type of the nth code word from all code words (such as DM QR, etc.)
Paramaters	n: Integer, range 0-N; the nth code word from all code words
Return value	String; the code type of the nth code word

Method	void get_exCodeType(int n)
Function Description	Retrieves the extended code type of the nth code word from all code words
Paramaters	n: Integer, range 0-N; the nth code word from all code words
Return value	String; the extended code type of the nth code word

Method	int get_msgLength(int n)
Function Description	Retrieves the length of the nth code word from all code words
Paramaters	n: Integer, range 0-N; the nth code word from all code words
Return value	Integer; the length of the nth code word

Method	void set_codes (int n)
Function Description	Sets or changes the total number of code words
Paramaters	n: Integer, total number of code words
Return value	none

Method	void set_decMsg (int n,string newMess,int length)
Function Description	Sets, edits, or modifies the nth code word
Paramaters	n: Integer, the nth code word from all code words newMess: String, new content of the code word length: Integer, length of the code word
Return value	None

Method	void set_codeType (int n,int type)
Function Description	Sets, edits, or modifies the code type of the nth code word
Paramaters	n: Integer, the nth code word from all code words

	type: Integer, code type of the code word
Return value	None

Method	void set_exCodeType (int n,int type)
Function Description	Sets, edits, or modifies the extended code type of the nth code word
Paramaters	n: Integer, the nth code word from all code words type: Integer, extended code type of the code word
Return value	None

Method	void set_ioEvent ()
Function Description	External IO control event
Paramaters	None
Return value	None

Method	int get_maxDecNum (void)
Function Description	Retrieves the maximum number of multi-code decodings
Paramaters	None
Return value	Number of multi-codes

Method	int get_minDecNum (void)
Function Description	Retrieves the minimum number of multi-code decodings
Paramaters	None
Return value	Number of multi-codes

Method	void set_result (int rv)
Function Description	Get the minimum decoding number of multi-codes
Paramaters	rv: 0: Indicates the entire script ran successfully 1: Indicates the entire script did not meet expectations
Return value	None

Method	string get_referenceCode (int index)
Function Description	Retrieves the reference barcode, supports setting predefined barcode content for comparison to check if the barcode content is correct.
Paramaters	Index: The index of the reference barcode
Return value	Content of the reference barcode

Javascript native common keywords

keywords	Description

var	Declares a variable. In JavaScript, var automatically matches the variable type. For example
	var mess = get_decMsg(i); // mess is of string type
	var count = get_coeds();// count is of integer type

if	judge
	var count = get_msgLength(0);
	if(count < = 10){
	// Executes if count is less than or equal to 10
	}else{
	// Executes if count is greater than 10
	}

for	Loop
	for (i = 0; i < get_coeds(); i++){
	// Executes the loop 10 times
	}

Introduction to Common String Processing Methods

Retrieve code word		var mess = get_decMsg(i)
Retrieve total number of code words		var count = get_codes();
substring(int start, int end)	Extracts characters between two specified index numbers in a string	var mess = get_decMsg(i);//12345 var tmp1 = mess .substring(0, 2); // Extracts the 0th and 1st bytes from mess
substr(int start,int length)	Extracts a specified number of characters from the start index	See use case 7
replace(string old, string new)	Replaces substrings matching a regex	var mess = get_decMsg(i); var temp = mess. replace('123', '456'); // Replaces '123' with '456' in the code word
toLowerCase()	Converts a string to lowercase	var mess= get_decMsg(i); mess = mess.toLowerCase();// Converts uppercase code word to lowercase; e.g., ABC -> abc
toLocaleLowerCase()	Converts a string to lowercase based on the host's locale, only a few languages (like Turkish) have locale-specific case mappings	
toLocaleUpperCase()	Converts a string to uppercase based on the host's locale, only a few languages (like Turkish) have locale-specific case mappings	
Retrieve code word length		var tmp = value.length; // Retrieves the code word length
split('').reverse().join('')	Reverses the order of a string	var mess= get_decMsg(i); var tmp = mess.split('').reverse().join('');//123456 -> 654321

lastIndexOf(string str)	Returns the position of the last occurrence of a specified character in a string	

indexOf(string str)	Returns the position of the first occurrence of a specified character in a string	var mess= get_decMsg(i);
		var index = mess.indexOf('1'); // Locates the position of character '1'
		var tmp = mess.substr(index + 1); // Extracts the code word after the index

search(string str)	Searches for a value matching a regex	See use case 18

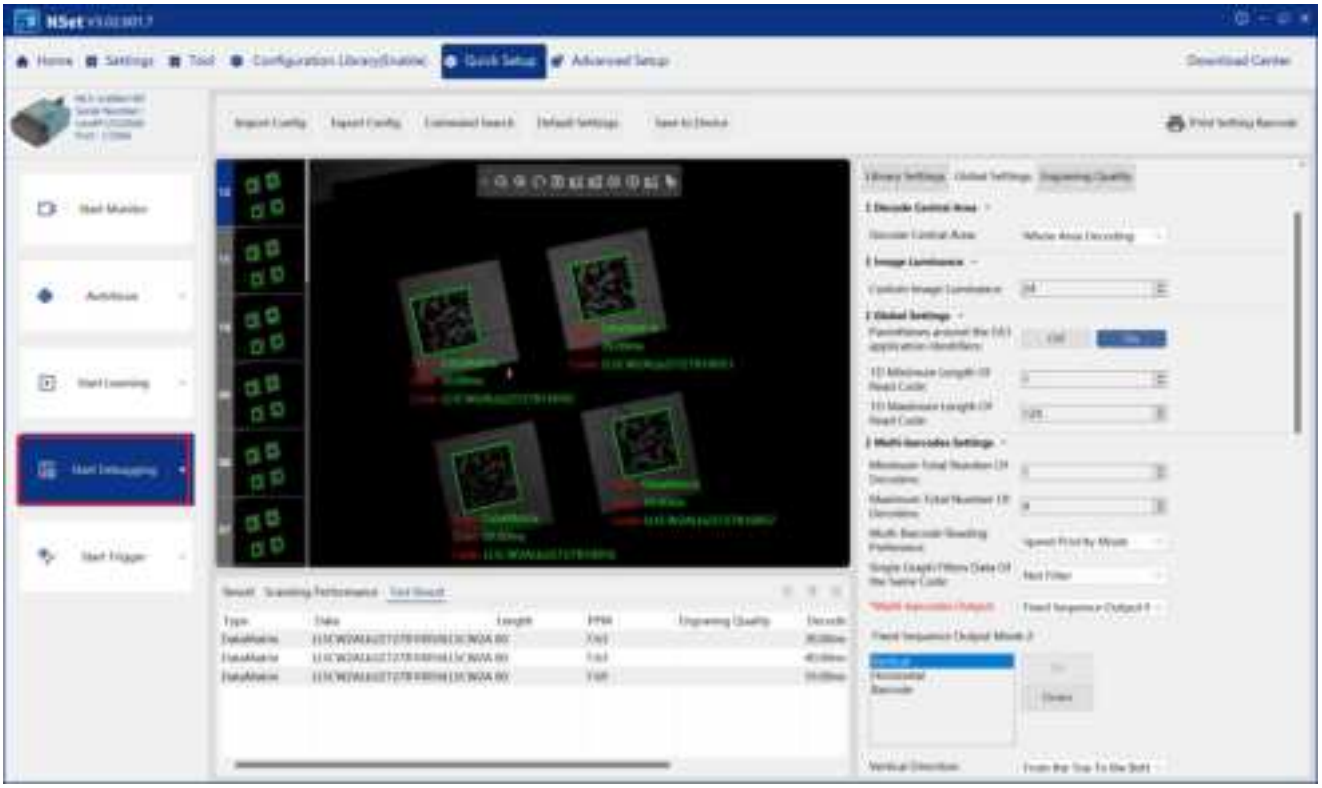
concat(string1， string2， string str3...)	Concatenates multiple strings into one	var count = get_codes();
		var messNew;
		for (i = 0; i < count; i++) {
		var mess= get_decMsg(i);
		mess = mess+'|';
		messNew.concat(mess);
		}

toString()	Typically used to convert a number to a string	var count = get_codes();// Retrieves the number of code words
		var mess = count.toString();// Converts the integer count to a string

7.13 Debugging

Start Debugging

The Start Debugging function is used to test the performance of configuration libraries. After learning a configuration library, you can perform debugging to validate and optimize it.



Start Trigger

The Start Trigger function is used to test the final performance of the decoding process. The trigger button simulates an external trigger input, enabling validation of the entire decoding process, including decoding modes, alternate configuration libraries, multi-barcode decoding, and other related features.

The statistical differences between debug mode and trigger decoding mode are as follows:

Parameter	Debug Mode	Trigger Decoding Mode
Test Content	Only tests the performance of the currently selected Library.	Tests the final application performance
Test Content	Number of decoded images	Number of trigger events
Success Count	Number of successfully decoded images	Number of successfully completed trigger events
Good Read Percent	Image decoding success rate (%)	Trigger event success rate (%)
Average Decode Time(ms)	Average image decoding time	Average decoding time for completed Trigger event (For multi-barcode reading across multiple images, this refers to the total decoding time for all barcodes across those images.)

The "Average Decode Time during Trigger Testing" has a special calculation method for multi-barcode reading across multiple images and continuous modes.

Scanning Mode	Good Read Percent	Average Decode Time
Standard Mode	Each decoding task completed within the trigger duration counts as one, even if multiple barcodes are read during a single trigger event in continuous mode.	Decoding time for the decoded image
Continuous Mode		Average decoding time within the trigger duration
Burst Mode		Decoding time for the decoded image

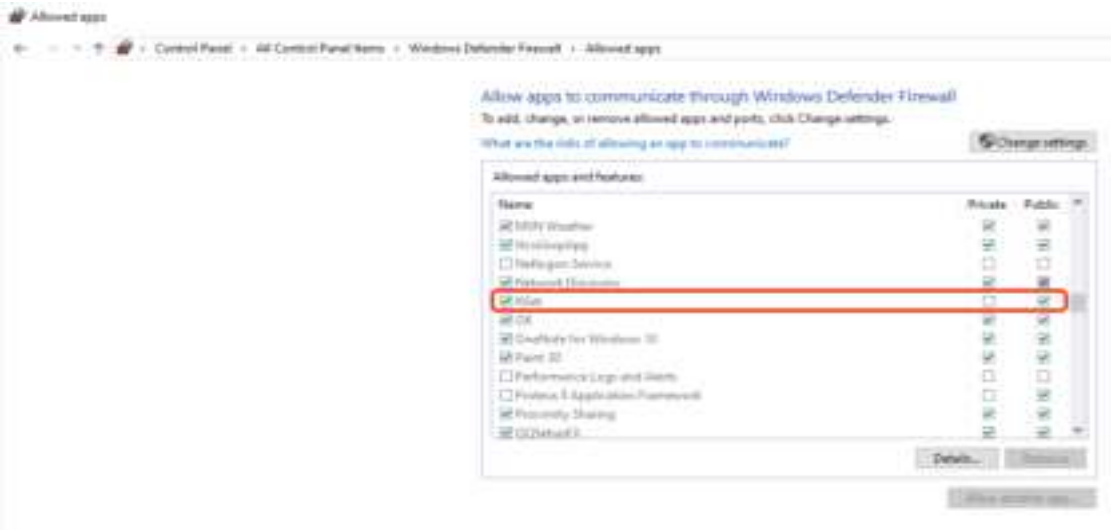
8 Troubleshootings

8.1 Device Not Found via Ethernet Connection

When using an Ethernet connection, if the device cannot be found on the homepage, please check the following:

1 Ensure the Ethernet cable is not damaged.

2 Ensure Windows Defender firewall allows communication with NSet.

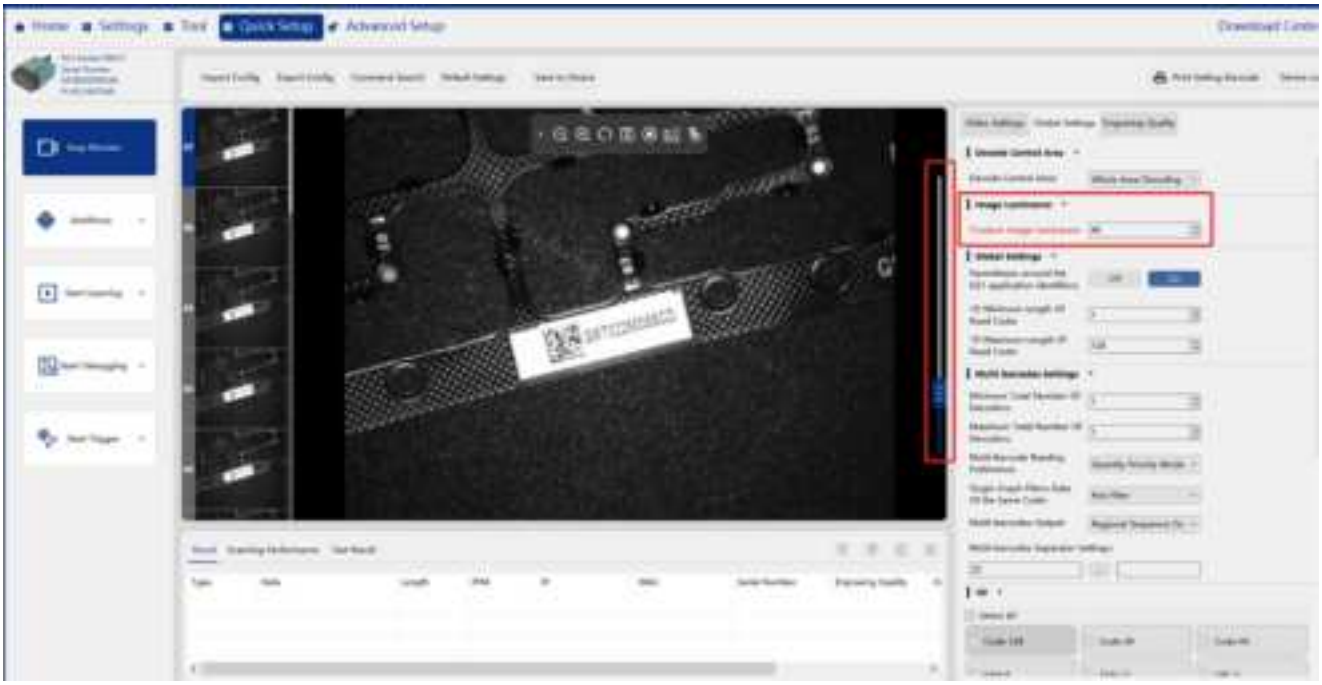


8.2 Focus Failure

There are two possible reasons for focus failure:

Ensure that the object to be focused is in the center of the image or laser aiming area, and there are no other objects (such as cables) in the center of the frame.

Perform the focusing after enabling the monitor, and ensure the brightness setting is appropriate. You can adjust the image brightness in Quick Setup > Global Settings > Image Luminance. If the image is too dark, increase the brightness; if overexposed, decrease it. Alternatively, open the monitor and manually adjust the brightness bar on the right side of the screen.

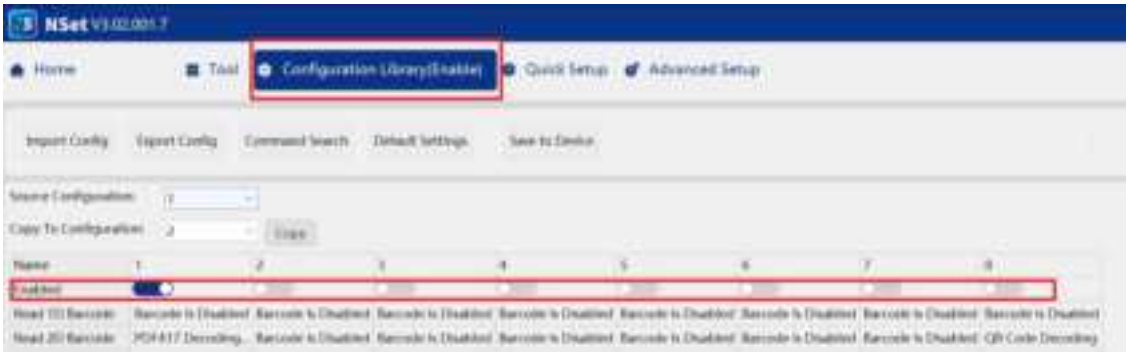


8.3 Unable to Decode a Different Barcode of the Same Symbology

After learning a barcode, the scanner may fail to read other barcodes of the same symbology on different objects. In this case, please re-learn the new barcode. Alternatively, go to the configuration library interface and re-enable the corresponding symbology for the current library.

8.4 No Image Displayed

Ensure that at least one configuration library is enabled, and the exposure time and gain settings are properly configured.



8.5 Device Group Configuration

Q1: How to output data via USB (or serial port) after device group configuration?
Connect the master device to the PC using a USB or serial cable. On the designated master device, enable Multi-barcode Data Output Channel, then enable either the USB Output Channel or RS232 Output Channel as per need.

Q2: Networking devices cannot be triggered simultaneously.
If devices in the network cannot be triggered at the same time, check the following:

Ensure that the Trigger Instruction is enabled on all devices in the network.

Delete and recreate the device group in NSet.

Ensure that the triggered device is set as the master.

Q3: ROI settings not found in NSet.
Go to Quick Setup > Global Settings > Decode Center Area，then select the Multiple ROI Decoding. The ROI Settings option will appear.

8.6 Fixed Barcode Position but Variable Location in Captured Images

The production line uses sensor triggering for the same objects at fixed positions, but the captured images show inconsistent left/right positioning.

Please disable the  image capture function or the Statistics monitoring feature.

The image capture function has limitations in scenarios with high timing requirements.

9 Appendix

9.1 Test Routines

Description	Input	Output	BarCode	Code

Extract the first 2 characters and the last 2 characters	NEWLAND 123	NE23		/**
				* Extract the first 2 characters and the last 2 characters, concatenate them
				*/
				var len = 0;
				var count = get_codes();
				for (i = 0; i < count; i++) {
				var code = get_decMsg(i);
				len = code.length;
				var tmp1 = code.substring(0, 2);
				var tmp2 = code.substring(len - 2, len);
				code = tmp1 + tmp2;
				set_codeType(i, i);
				set_exCodeType(i, i);
				set_decMsg(i, code, code.length);
				print('code = %s', code);
				}

Delete the third character from the data read	NEWLAND 123	NELAND123		/**
				* Remove the third character
				*/
				var len = 0;
				var count = get_codes();
				
				for (i = 0; i < count; i++) {
				var code = get_decMsg(i);
				len = code.length;
				var tmp1 = code.substring(0, 2);

				var tmp2 = code.substring(3, len);
				code = tmp1 + tmp2;
				set_codeType(i, i);
				set_exCodeType(i, i);
				set_decMsg(i, code, code.length);
				print('code = %s', code);
				}

Add character '65' to the end of the read data	NEWLAND123	NEWLAND12365		/**
				* Add '65' at the end
				*/
				var count = get_codes();
				
				for (i = 0; i < count; i++) {
				var code = get_decMsg(i);
				code = code + '65';
				set_codeType(i, i);
				set_exCodeType(i, i);
				set_decMsg(i, code, code.length);
				print('code = %s', code);
				}

Remove leading zeros from the barcode	00000NEWLAND	NEWLAND		/**
				* Remove all leading zeros
				* Use a regular expression to find leading zeros
				*/
				
				var pattr = /^[0]*/;
				var count = get_codes();
				
				for (i = 0; i < count; i++) {
				var code = get_decMsg(i);
				code = code.replace(pattr, '');
				set_codeType(i, i);
				set_exCodeType(i, i);
				set_decMsg(i, code, code.length);
				print('code = %s', code);
				}

Convert all characters to lowercase	NEWLAND	newland		/**
				* Convert all uppercase letters to lowercase
				*/
				var count = get_codes();
				
				for (i = 0; i < count; i++) {
				var code = get_decMsg(i);
				code = code.toLowerCase();
				set_codeType(i, i);
				set_exCodeType(i, i);
				set_decMsg(i, code, code.length);
				print('code = %s', code);
				}

Read the length of the data	NEWLAND	7		/**
				* Get the length of the string
				*/
				var count = get_codes();
				
				/**
				* Output all barcodes starting with 1
				*/
				function strfunc(value) {
				var tmp = value.length;
				return tmp.toString();
				}
				
				for (i = 0; i < count; i++) {
				var code = get_decMsg(i);
				code = strfunc(code);
				set_codeType(i, i);
				set_exCodeType(i, i);
				set_decMsg(i, code, code.length);
				print('code = %s', code);
				}

Read characters from the 2nd to the 5th position	NEWLAND	EWLAN		/**
				* Read characters from position 2 to 5
				*/
				var count = get_codes();
				
				for (i = 0; i < count; i++) {
				var code = get_decMsg(i);
				code = code.substr(1, 5);
				set_codeType(i, i);
				set_exCodeType(i, i);
				set_decMsg(i, code, code.length);
				print('code = %s', code);
				}

Replace '123' in the string with '456'	NEWLAND 123	NEWLAND456		/**
				* Replace '123' in the string with '456'
				*/
				var count = get_codes();
				
				for (i = 0; i < count; i++) {
				var code = get_decMsg(i);
				code = code.replace('123', '456');
				set_codeType(i, i);
				set_exCodeType(i, i);
				set_decMsg(i, code, code.length);
				print('code = %s', code);
				}

Reverse the order of the string	NEWLAND 123	321DNALWEN		/**
				* Reverse the order of the string
				*/
				function strfunc(value) {
				var tmp = value.split('').reverse().join('');
				return tmp;
				}
				
				var count = get_codes();

				for (i = 0; i < count; i++) {
				  var code = get_decMsg(i);
				  /* */
				  code = strfunc(code);
				  /* */
				  set_codeType(i, i);
				  set_exCodeType(i, i);
				  set_decMsg(i, code, code.length);
				  print('code = %s', code);
				}

Remove characters 'E' and 'A' from the data	NEWLAND 123	NWLND123		/**
				 * Remove characters 'E' and 'A' from the data
				 */
				var pattr = /[A|E]/g;
				
				function strfunc(value) {
				  var tmp = value.replace(pattr, '');
				  return tmp;
				}
				
				var count = get_codes();
				for (i = 0; i < count; i++) {
				  var code = get_decMsg(i);
				
				  code = strfunc(code);
				
				  set_codeType(i, i);
				  set_exCodeType(i, i);
				  set_decMsg(i, code, code.length);
				  print('code = %s', code);
				}

Return the 5th character of the data	NEWLAND 123	A		/**
				 * Return the 5th character of the data
				 */
				function strfunc(value) {
				  var tmp = value.charAt(4);
				  return tmp;
				}
				
				var count = get_codes();

				for (i = 0; i < count; i++) {
				  var code = get_decMsg(i);
				
				  code = strfunc(code);
				
				  set_codeType(i, i);
				  set_exCodeType(i, i);
				  set_decMsg(i, code, code.length);
				  print('code = %s', code);
				}

Replace all '|' with '#'	123|240|2021|123456	123#240#2021#123456		/**
				 * Replace all '|' with '#'
				 */
				function strfunc(value) {
				  var pattr = /[|]/g;
				  var tmp = value.replace(pattr, '#');
				  return tmp;
				}
				
				var count = get_codes();
				for (i = 0; i < count; i++) {
				  var code = get_decMsg(i);
				
				  code = strfunc(code);
				
				  set_codeType(i, i);
				  set_exCodeType(i, i);
				  set_decMsg(i, code, code.length);
				  print('code = %s', code);
				}

Pad the data with zeros up to 15 characters	NEWLAND	NEWLAND00000000		/**
				 * Pad the data with zeros up to the 15th character
				 */
				function strfunc(value) {
				  var tmp = value;
				  var remind = 15;
				
				  for (var i = value.length; i < remind; i ++) {

				tmp = tmp + '0';
				}
				return tmp;
				}
				
				var count = get_codes();
				for (i = 0; i < count; i++) {
				var code = get_decMsg(i);
				
				code = strfunc(code);
				
				set_codeType(i, i);
				set_exCodeType(i, i);
				set_decMsg(i, code, code.length);
				print('code = %s', code);
				}

Remove the first and last character, keeping the middle part	NEWLAND 123	EWLAND12		/**
				* Remove the first and last character, keeping the middle part
				*/
				function strfunc(value) {
				var len = value.length;
				
				tmp = value.substr(1, len - 2);
				
				return tmp;
				}
				
				var count = get_codes();
				for (i = 0; i < count; i++) {
				var code = get_decMsg(i);
				
				code = strfunc(code);
				
				set_codeType(i, i);
				set_exCodeType(i, i);
				set_decMsg(i, code, code.length);
				print('code = %s', code);
				}

Output all characters before '1'	NEWLAND
	123
		NEWLAND		/**
				* Output all characters before '1'
				*/
				function strfunc(value) {
				var len = value.length;
				var index = value.indexOf('1');
				tmp = value.substr(0, index);
				
				return tmp;
				}
				
				var count = get_codes();
				for (i = 0; i < count; i++) {
				var code = get_decMsg(i);
				
				code = strfunc(code);
				
				set_codeType(i, i);
				set_exCodeType(i, i);
				set_decMsg(i, code, code.length);
				print('code = %s', code);
				}

Output all characters after '1'	NEWLAND
	123
		23		/**
				* Output all characters after '1'
				*/
				function strfunc(value) {
				var len = value.length;
				var index = value.indexOf('1');
				tmp = value.substr(index + 1);
				
				return tmp;
				}
				
				var count = get_codes();
				for (i = 0; i < count; i++) {
				var code = get_decMsg(i);
				
				code = strfunc(code);
				
				set_codeType(i, i);
				set_exCodeType(i, i);
				set_decMsg(i, code, code.length);

				print('code = %s', code); }

If the data starts with '1', output 'A'; otherwise, output 'B'	1TT
	2TT
		A
		B
			
			1TT
			 
			2TT
				/**
				* If the data starts with '1', output 'A'; otherwise, output 'B'
				*/
				function strfunc(value) {
				var len = value.length;
				var tmp = value.substr(0, 1);
				if (tmp == '1') {
				tmp = 'A';
				} else if (tmp == '2') {
				tmp = 'B';
				}
				
				return tmp;
				}
				
				var count = get_codes();
				for (i = 0; i < count; i++) {
				var code = get_decMsg(i);
				
				code = strfunc(code);
				
				set_codeType(i, i);
				set_exCodeType(i, i);
				set_decMsg(i, code, code.length);
				print('code = %s', code);
				}

Append "_OK" to the end of the data if it contains 456; otherwise, append "_NG"	1TT456
	2TT
		1TT456_OK
		2TT_NG
			
			1TT456
				/**
				* Append "_OK" to the end of the data if it contains 456; otherwise, append "_NG"
				*/
				
				function strfunc(value) {
				var len = value.length;
				var pattr = /456/;
				var pos = value.search(pattr);

			2TT	if (pos >= 0) {
				tmp = value + "_OK";
				} else {
				tmp = value + "_NG";
				}
				
				return tmp;
				}
				
				var count = get_codes();
				for (i = 0; i < count; i++) {
				var code = get_decMsg(i);
				
				code = strfunc(code);
				
				set_codeType(i, i);
				set_exCodeType(i, i);
				set_decMsg(i, code, code.length);
				print('code = %s', code);
				}

Output "OK" if the data contains control characters; otherwise, output "NG"	1TT456
	234<HT>12
	3
		NG
		OK
				/**
				* Output "OK" if the data contains control characters; otherwise, output "NG"
				*/
				function strfunc(value) {
				var len = value.length;
				var pattr = /\s+/g;
				var index = value.search(pattr);
				var tmp = value;
				
				if (index > 0) {
				tmp = 'OK';
				} else {
				tmp = 'NG';
				}
				return tmp;
				}
				
				var count = get_codes();
				for (i = 0; i < count; i++) {
				var code = get_decMsg(i);

				code = strfunc(code);
				
				set_codeType(i, i);
				set_exCodeType(i, i);
				set_decMsg(i, code, code.length);
				print('code = %s', code);
				}

Remove control characters from the data	234<HT>123	234123		/**
				* Remove control characters from the data
				*/
				function strfunc(value) {
				var len = value.length;
				var pattr = /\s+/g;
				var tmp = value.replace(pattr, '');
				return tmp;
				}
				
				var count = get_codes();
				for (i = 0; i < count; i++) {
				var code = get_decMsg(i);
				
				code = strfunc(code);
				
				set_codeType(i, i);
				set_exCodeType(i, i);
				set_decMsg(i, code, code.length);
				print('code = %s', code);
				}

Move 'a' three positions to the right	newland123	newlnd1a23		/**
				* Move 'a' three positions to the right
				*/
				function strfunc(value) {
				var len = value.length;
				var index = value.indexOf('a');
				var tmp = value;
				var tmp1 = value;
				var tmp2 = value;

				if (index >= 0) {
				var tmp = value.substr(0, index);
				var tmp1 = value.substr(index + 1
				, 3);
				var tmp2 = value.substr(index + 4
				, len);
				tmp = tmp + tmp1 + 'a' + tmp2;
				}
				return tmp;
				}
				
				var count = get_codes();
				for (i = 0; i < count; i++) {
				var code = get_decMsg(i);
				
				code = strfunc(code);
				
				set_codeType(i, i);
				set_exCodeType(i, i);
				set_decMsg(i, code, code.length);
				}

Concatenate multiple barcodes with separators and output as a string	NEWLAND
	123
	456
	abC
		NEWLAND,123,456,abC		/**
				* Concatenate multiple barcodes with separators and output as a string
				*/
				function strfunc(value) {
				var tmp = '';
				for (var i = 0; i < value.length; i++) {
				if(value[i].length != 0){
				tmp = tmp + value[i] + ',';
				}
				}
				
				tmp = tmp.substr(0, tmp.length - 1);
				return tmp;
				}
				
				var codes = new Array();
				var count = get_codes();
				for (var i = 0; i < count; i++) {
				codes[i] = get_decMsg(i);

				set_decMsg(i, '', 0);
				if (i == count - 1) {
				var out = strfunc(codes);
				set_decMsg(0, out, out.length);
				set_codes(1);
				}
				
				}

Insert "LTT" between the second and third data read	NEWLAND
	123
	456
	abC
		NEWLAND123LTT
		456abC
				/**
				* Insert "LTT" between the second and third data read
				*/
				function strfunc(value) {
				var tmp = '';
				for (var i = 0; i < value.length; i++) {
				if (i == 2) {
				tmp = tmp + "LTT";
				}
				
				if (value[i].length != 0) {
				tmp = tmp + value[i];
				}
				}
				
				return tmp;
				}
				
				var codes = new Array();
				var count = get_codes();
				for (var i = 0; i < count; i++) {
				codes[i] = get_decMsg(i);
				set_decMsg(i, '', 0);
				if (i == count - 1) {
				var out = strfunc(codes);
				set_decMsg(0, out, out.length);
				set_codes(1);
				}
				
				}

Delete all barcodes starting with '1' and output	NEWLAND
	123
	456
	abC
		NEWLAND456abC		/**
				* Delete all barcodes starting with '1' and output
				*/
				function strfunc(value) {
				var tmp = '';
				
				for (var i = 0; i < value.length; i++) {
				var index = value[i].search('1');
				if (index >= 0) {
				delete value[i];
				} else if(value[i].length > 0){
				tmp = tmp + value[i];
				}
				
				}
				return tmp;
				}
				
				var count = get_codes();
				var newCodes = new Array();
				for (i = 0; i < count; i++) {
				newCodes[i] = get_decMsg(i);
				if (i == count - 1) {
				var out = strfunc(newCodes);
				set_decMsg(0, out, out.length);
				set_codes(1);
				}
				}

Only output barcodes starting with 'F'	NEWLAND
	123
	F15
	456
	abC
	F12
		F12F15		/**
				* Only output barcodes starting with 'F'
				*/
				function strfunc(value) {
				var tmp = '';
				
				for (var i = value.length - 1; i >= 0; i--) {
				var index = value[i].search('F');
				if (index == 0) {
				tmp = tmp + value[i];

				} else {
				delete value[i];
				}
				
				}
				return tmp;
				}
				
				var count = get_codes();
				var newCodes = new Array();
				for (i = 0; i < count; i++) {
				newCodes[i] = get_decMsg(i);
				if (i == count - 1) {
				var out = strfunc(newCodes);
				set_decMsg(0, out, out.length);
				set_codes(1);
				}
				}

Sort by numerical size and letter order	A12
	B12
	C21
	A22
	B13
		A12A22B12B13C21		/**
				* Sort by numerical size and letter order
				*/
				function strfunc(value) {
				var tmp = '';
				
				value.sort();
				for (var i = 0; i < value.length; i++) {
				tmp = tmp + value[i];
				}
				return tmp;
				}
				
				var count = get_codes();
				var newCodes = new Array();
				for (i = 0; i < count; i++) {
				newCodes[i] = get_decMsg(i);
				if (i == count - 1) {
				var out = strfunc(newCodes);
				set_decMsg(0, out, out.length);
				set_codes(1);
				}

				}

Place the data 'A12' in the fifth position	A12
	B12
	C21
	A22
	B13
		B12C21A22B13A12	
			 
			 
			 
			 
				/**
				* Place the data 'A12' in the fifth position
				*/
				function strfunc(value) {
				var tmp = '';
				var tmp1 = '';
				
				for (var i = 0; i < value.length; i++) {
				if (value[i] == 'A12') {
				tmp1 = value[i];
				delete value[i];
				} else {
				tmp = tmp + value[i];
				}
				}
				tmp = tmp + tmp1;
				return tmp;
				}
				
				var count = get_codes();
				var newCodes = new Array();
				for (i = 0; i < count; i++) {
				newCodes[i] = get_decMsg(i);
				if (i == count - 1) {
				var out = strfunc(newCodes);
				set_decMsg(0, out, out.length);
				set_codes(1);
				}
				}

Set barcode expected content as tin*	Input:
	- tin1
	- LTT
	- tin232
		Output:
		- tin1
		- tin232
				/**
				* Replace tin*
				*/
				function strfunc(value) {
				var tmp = '';
				for (var i = 0; i < value.length; i++) {

			
			 
				var index = value[i].search('tin');
				if (index < 0) {
				delete value[i];
				} else if(value[i].length > 0){
				tmp = tmp + value[i];
				}
				}
				return tmp;
				}
				
				//*
				var count = get_codes();
				var newCodes = new Array();
				for (i = 0; i < count; i++) {
				newCodes[i] = get_decMsg(i);
				if (i == count - 1) {
				var out = strfunc(newCodes);
				set_decMsg(0, out, out.length);
				set_codes(1);
				}
				}

Set barcode expected content as tin??	Input: - tin11 - LTT - tin232	Output: - tin11	  	

Exclude content 123	Input:**
	- 123
	- 345
	- 678
		Output:
		- 345
		- 678
				/**
				* Exclude 123*
				*/
				function strfunc(value) {
				var tmp = '';
				for (var i = 0; i < value.length; i++) {
				var index = value[i].search('123');
				if (index >= 0) {
				delete value[i];
				} else if (value[i].length > 0) {
				tmp = tmp + value[i];
				}
				}
				return tmp;
				}
				//*
				var count = get_codes();
				var newCodes = new Array();
				for (i = 0; i < count; i++) {
				newCodes[i] = get_decMsg(i);
				if (i == count - 1) {
				var out = strfunc(newCodes);
				set_decMsg(0, out, out.length);
				set_codes(1);
				}
				}

Attach date and time to read data	123456	2021/02/24 12:31 123456		/**
				* Attach date and time to read data
				* Read data: 123456
				* Execution result: 2021/02/24 12:31 123456
				*/
				function strfunc(value) {
				var tmp = '';
				var date = new Date();
				tmp = date.getFullYear().toString() + '/';
				tmp = tmp + date.getMonth().toString() + '/';
				tmp = tmp + date.getDate().toString() + ' ';
				tmp = tmp + date.getHours().toString() + ':';
				tmp = tmp + date.getMinutes().toString() + ' ';
				tmp = tmp + value;
				return tmp;
				}
				
				var count = get_codes();
				var newCodes = new Array();
				for (i = 0; i < count; i++) {
				newCodes[i] = get_decMsg(i);
				if (i == count - 1) {
				var out = strfunc(newCodes);
				set_decMsg(0, out, out.length);
				set_codes(1);
				}
				}

Determine if the date in the data exceeds June 1, 2021. If not exceeded, output OK, otherwise output NG	20210301	OK/NG	 20210301	/** * Determine if the date in the data exceeds June 1, 2021. If not exceeded, output OK, otherwise output NG */ function strfunc(value) { var tmp = '';

			
			20160301
				var today = new Date();
				today.setFullYear(2021, 6, 1);
				var year = parseInt(value.slice(0, 4));
				var month = parseInt(value.slice(4, 6));
				var day = parseInt(value.slice(7));
				var messDay = new Date();
				messDay.setFullYear(year, month, day);
				if (messDay > today) {
				tmp = 'OK';
				} else {
				tmp = 'NG';
				}
				return tmp;
				}
				var count = get_codes();
				var newCodes = new Array();
				for (i = 0; i < count; i++) {
				newCodes[i] = get_decMsg(i);
				if (i == count - 1) {
				var out = strfunc(newCodes[i]);
				set_decMsg(0, out, out.length);
				set_codes(1);
				}
				}

9.2 Quick Settings Barcodes

Scanner Information

[Query All Information]

Default Settings

[Restore All Factory Defaults]

[Restore the Factory Defaults of Ethernet]

Scan Mode

[Continuous Mode]

[Standard Mode]

[Burst Mode]

Trigger Mode

[Level Trigger Mode]

[Pulse Mode]

Input IO Level Control

【Set to Active High for Input 1】

【Set to Active Low for Input 1】

【Set to Active High for Input 2】

【Set to Active Low for Input 2】

Aimer Settings

@AMLENA0

【Disable】

@AMLENA1

【Enable】

@AMLENA2

【Always On】

Terminator Suffix

@TSUEAN0

【Disable】

@TSUENA1.TSUSET0D

【Set to CR(0X0D)】

@TSUENA1.TSUSET0D0A

【Set to CRLF(0X0D,0X0A)】

Stay Connect with us @Newland AIDC

Newland AIDC

No.1 Rujiang West Rd., Mawei, Fuzhou, Fujian 350015, China
+86-591-83979500 info@newlandaidc.com www.newlandaidc.com

Asia Pacific

Add: 8 Raffles Quay #14-08 Singapore 048583
Tel: +86 591 83979500
Email: info@newlandaidc.com

Taiwan, China
Add: 7F-6, No. 268, Liancheng Rd., Zhonghe Dist. 235, New Taipei City, Taiwan
Tel: +886 2 7731 5388

Japan
住所: 〒108-0075
[illegible]
[illegible]
電話: +84 03 4405 3222

Korea
Add: B2, Center Best-one, Jong-eun Medical Plaza 5F, Bojeong-dong 1261-4, Kihung-gu, Yongin-City, Kyunggi-do, South Korea
Tel: +82 10 8990 4838

India
Add: Office no. 309-311, 3rd Floor, Tower B, NOIDA ONE business park B 8, Block B, Industrial Area, Sector 62, Noida, Uttar Pradesh 201309
Tel: +91-120-3201449 /50 /51 /52

Indonesia
Add: Eightyeight@kasablanka Tower A 12th Floor Unit A&H, Jl. Casablanca Raya Kav. 88, Jakarta Selatan 12870
Tel: +62 21 3950 5400

Vietnam
Tel: +84 969712692

Malaysia
Tel: +60 322042828

Thailand
Tel: +66 871485745

Europe & Middle East & Africa

Add: Rolweg 25, 4104 AV Culemborg, The Netherlands
Tel: +31 (0) 345 87 00 33 Web: www.newland-id.com
Email: sales@newland-id.com Tech Support: tech-support@newland-id.com

North America

Add: 46559 Fremont Blvd., Fremont, CA 94538, USA
Tel: +1 510 490 3888
Email: info@newlandaidc.com

Latin America

Tel: +1 239 598 0088
Email: info@newlandaidc.com

Chile
Tel: +56 9 9337 3177

Central America & Caribbean
Tel: +52 155 5432 9079

Brazil
Tel: +55 35 9767 8078

Colombia
Tel: +57 310 387 4484

Mexico
Tel: +(001) 323 443-2570

Newland AIDC
Scanning Made Simple