RICOH

# RICOH SC-20

# Socket Mode Function Operating Instructions

# How to Read This Manual

## About Symbols

The symbols used in this manual have the following meanings.

**Important**

Indicates an explanation containing points to pay attention to when operating the device, restrictions, or other information. Be sure to read the explanation.

**Note**

Indicates an explanation containing information that is useful to know, a supplementary operating procedure, or other information.

**Reference** /(→P. ##)

Indicates reference information.

[ ]

Indicates a screen item or button name.

# Contents

# 1. Overview

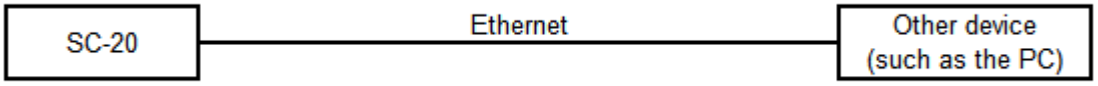
The SC-20 can be connected to an external device by using the Socket Mode function of TCP/IP.
This manual describes the socket mode connection procedure and the data format to be set during the Socket Mode.

## Connection Configuration

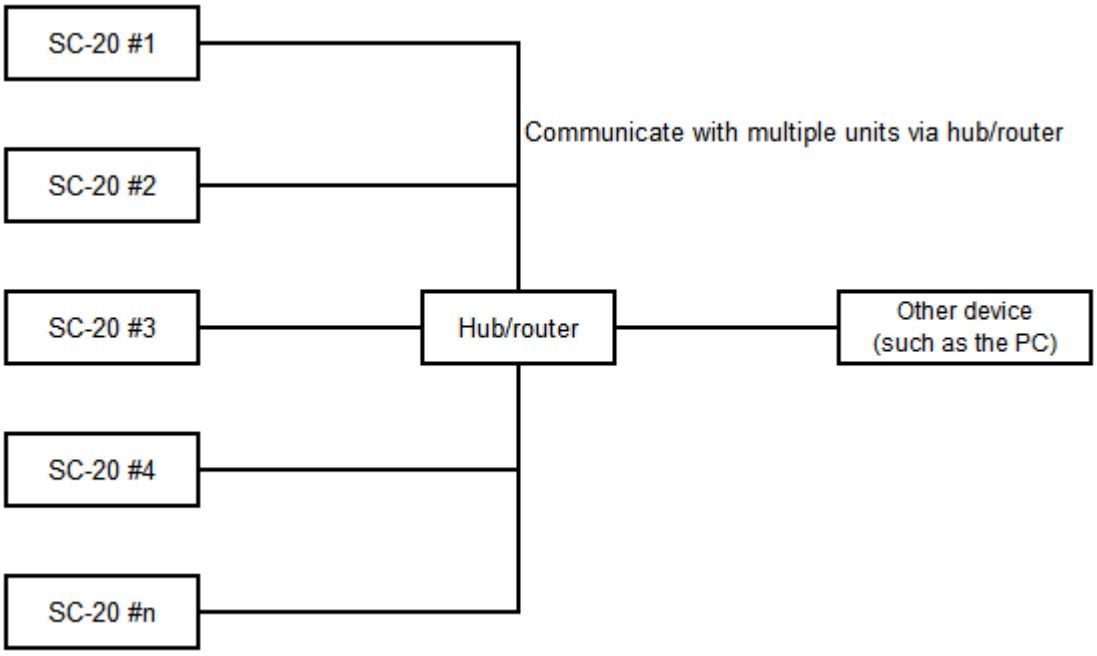
### Connection configuration

If you use an Ethernet connection, you can connect multiple inspection camera systems as shown in the figure below.
SC-20 operates as a client. Therefore, it is necessary to construct server software for operation in the other device.

Example 1:




Example 2:



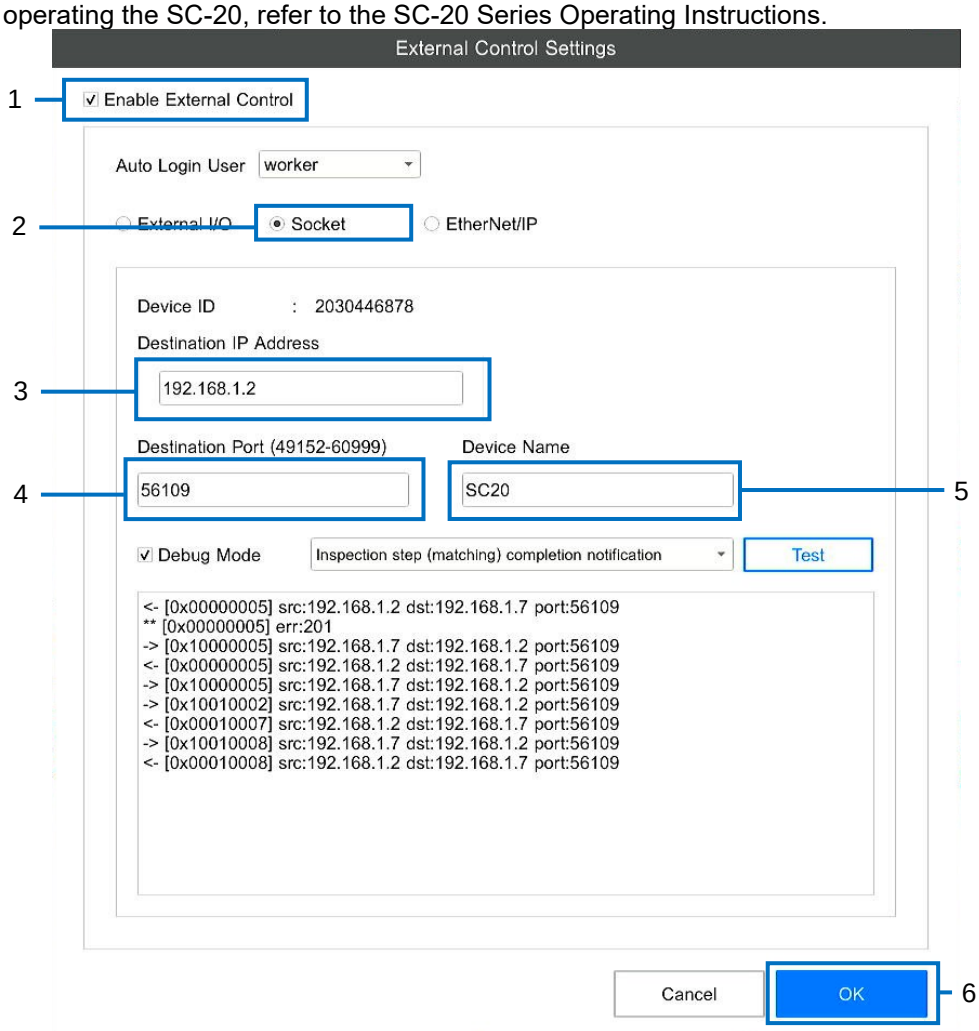
# 2. Socket Mode

## Enabling the Socket Mode Control Function

Log in to the SC-20 in the “Administrator Mode”, and select [External Control Settings ...] from the [System Settings] menu to view the screen shown in the figure below.

**Reference**

- For details on operating the SC-20, refer to the SC-20 Series Operating Instructions.




1. **Select the [Enable External Control] checkbox.**
2. **Select [Socket Mode Function].**
3. **Set the IP address of the other device (socket mode destination) in [Destination IP Address].**
4. **Set the port number of the other device in [Destination Port].**
5. **Enter the name of the device in [Device Name].**
   Enter any arbitrary name in 1 to 50 single-byte alphanumeric characters.
6. **Click [OK].**
   The settings are saved, and the function is reflected after the device restarts.

**Note**

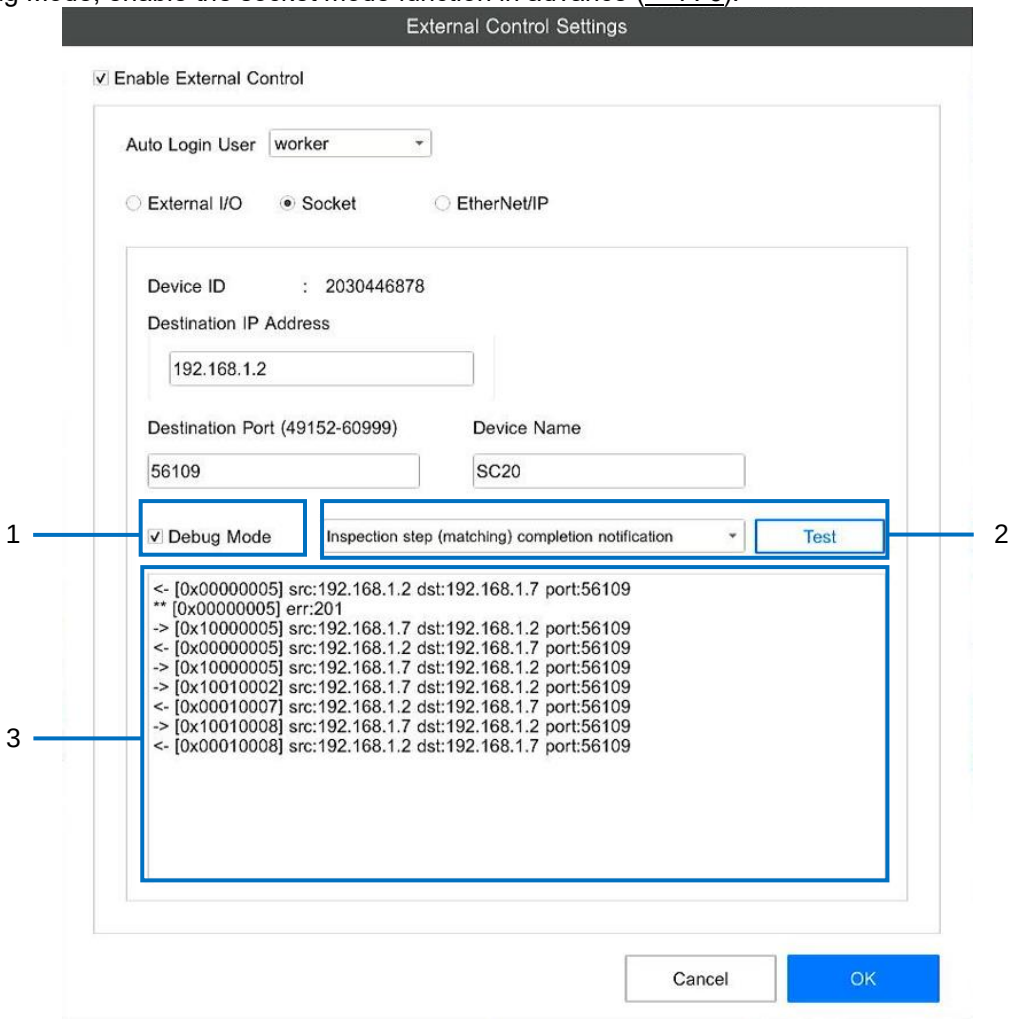
- The [Device ID] is automatically set by the system.

# Configuring the Debug Mode

You can set the Debug Mode. During the development of applications of the other device, you can perform a connectivity test and operation verification by using the Debug Mode.

**Note**

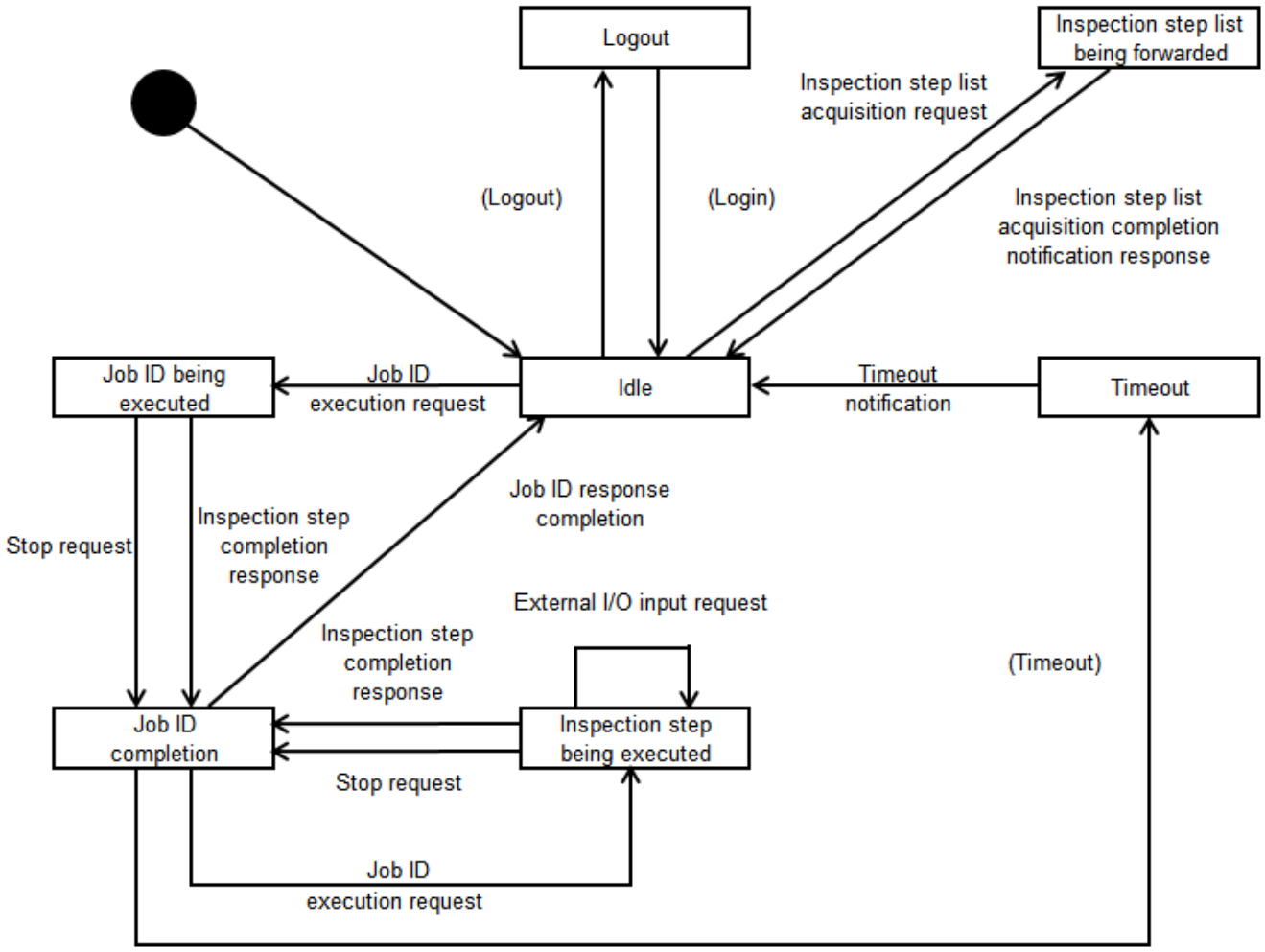
- To set the Debug Mode, enable the socket mode function in advance (→ P. 6).




1. **Place a check mark on [Debug Mode].**
2. **From the pull-down on the left, select a notification message and click [Test].**
   It is possible to send a notification message containing dummy data according to the pull-down menu.
3. **The message log of the SC-20 and the other device is displayed.**

## Commands and System Status

The commands exchanged in the Socket Mode and the status transition of the system is described below.
The processes enclosed within parentheses correspond to operations either performed manually by the user or automatically within the system.



# Sequence

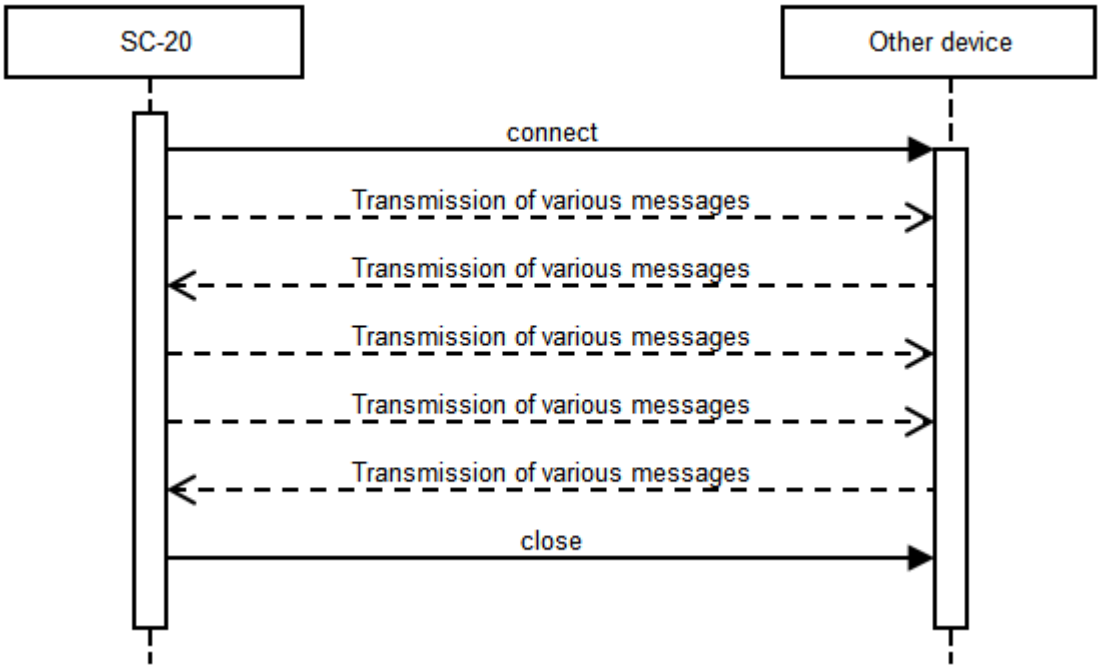
The sequences assumed in the Socket Mode control function are described below.

**Important**

- If communication is performed outside the sequences shown below, it will not be possible to perform the correct processing.
- Do not perform request command transmission within the sequence of another command.

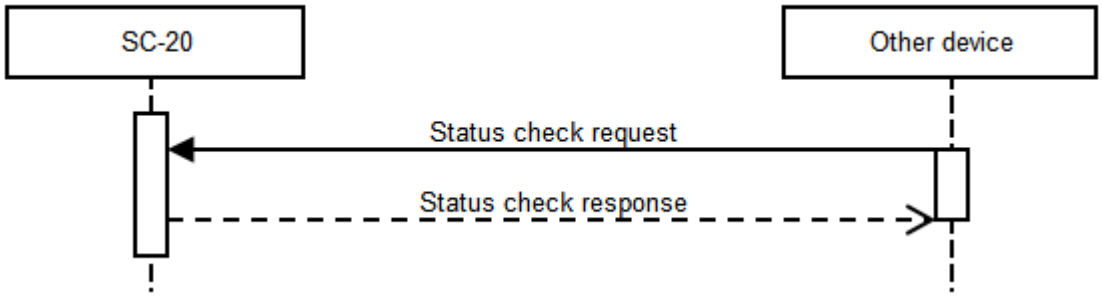
## Connection method

The connection method of Socket Mode is the client mode only.
This connection method is applicable when “connect” is performed and continues to be maintained until system outage has ended.



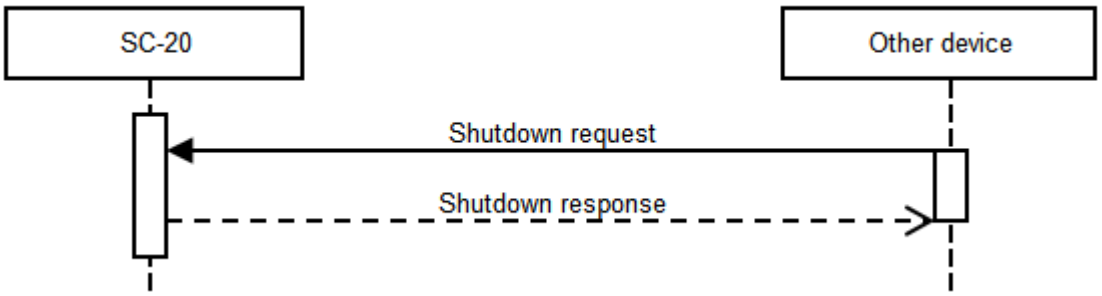

## Status check

To check the status of the SC-20 from the other device, send a “Status check request” from the other device.
The camera status is sent as “Status check response” at any arbitrary timing.



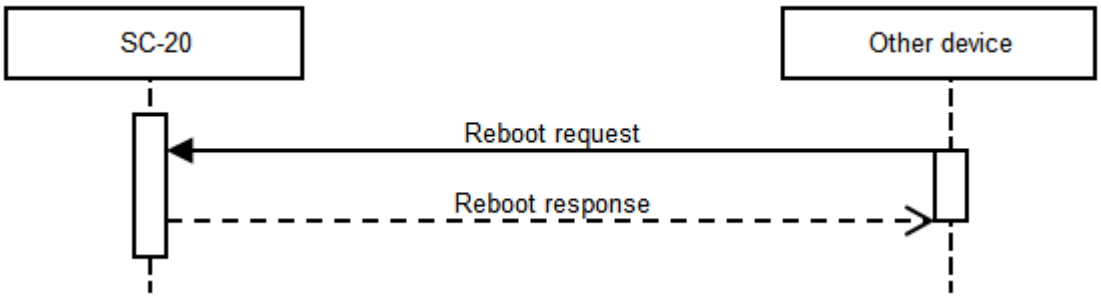
## Shutdown

To shut down the SC-20 from the other device, send a “Shutdown request” from the other device.
The camera status is sent as “Shutdown response” at any arbitrary timing.



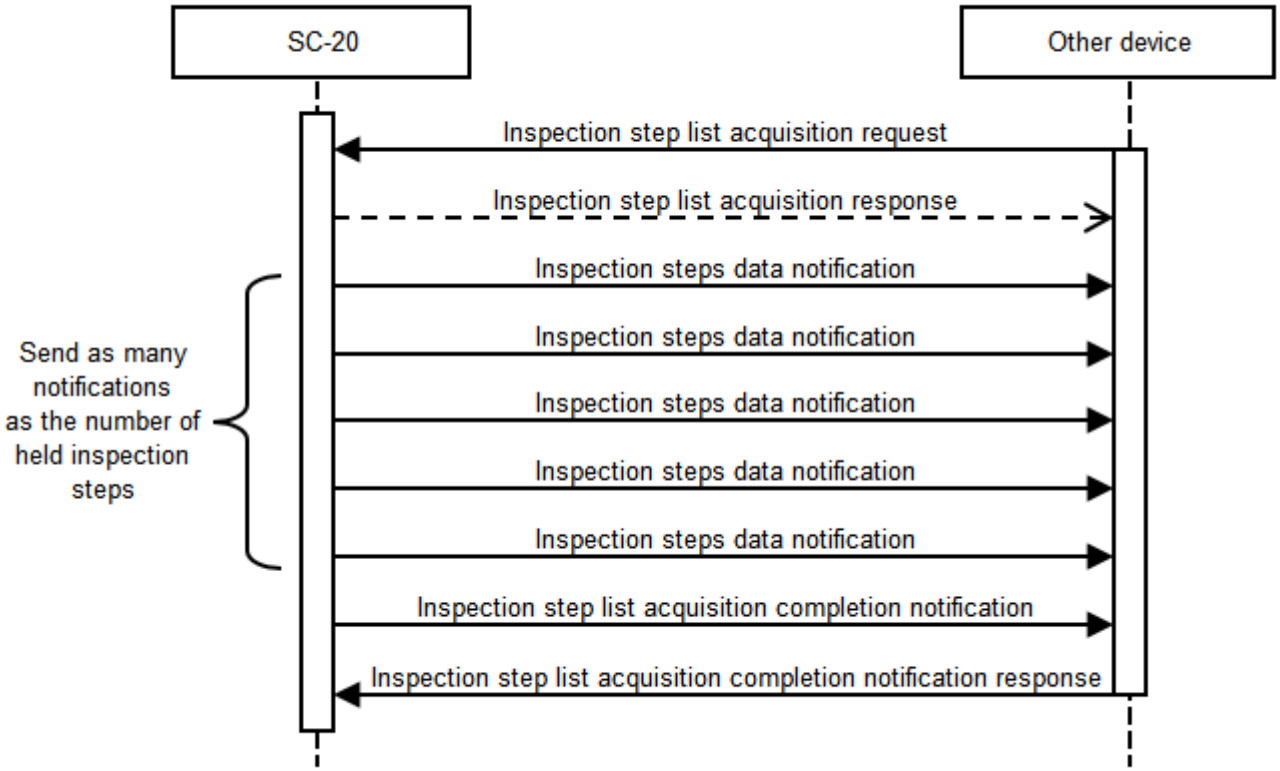

## Reboot

To restart the SC-20 from the other device, send a “Reboot request” from the other device.
The camera status is sent as “Reboot response” at any arbitrary timing.



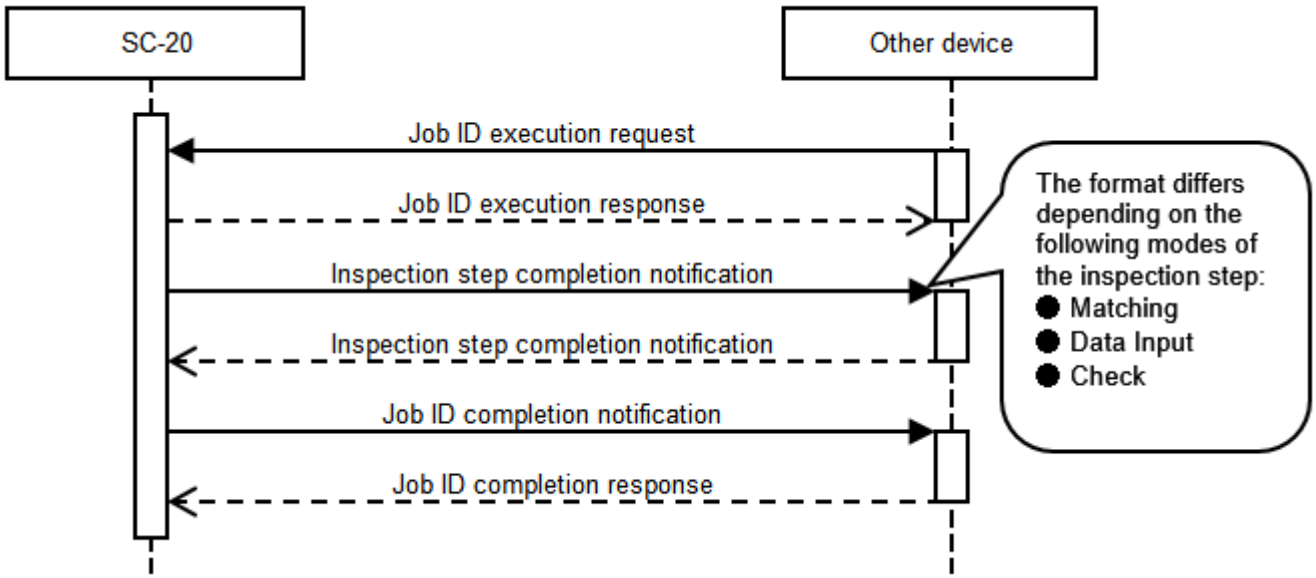
## Acquiring inspection step list

To acquire an inspection step registered in the SC-20, send an “Inspection step list acquisition request” from the other device. After sending an “Inspection steps response”, the SC-20 sends an “Inspection steps data notification”.
The “Inspection steps data notification” is a single-time notification for a single inspection step registered in the SC-20. Once the transmission of the “Inspection steps data notification” is complete for all inspection steps, send an “Inspection step list acquisition completion notification”. The other device sends an “Inspection step list acquisition completion response” and the sequence ends.



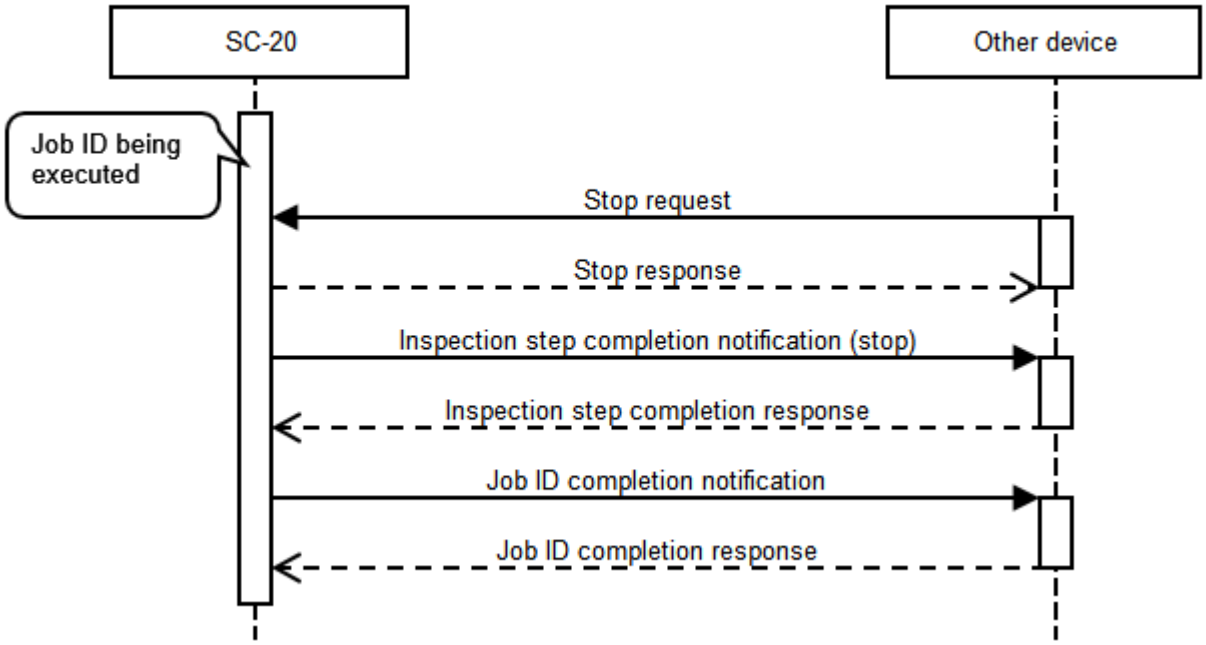
## Job ID execution processing

To switch a job ID of the SC-20 from the other device and execute a sequence from the first inspection step registered in the job ID, send a “Job ID execution request” from the other device. Sequential execution is carried out from the first inspection step registered in the job ID. The execution result is sent from the SC-20 as an “Inspection step completion notification”.
Once the execution of all inspection steps registered in the job ID is complete, the SC-20 sends a “Job ID completion notification”. The other device sends a “Job ID completion response” to the SC-20 and the sequence ends.



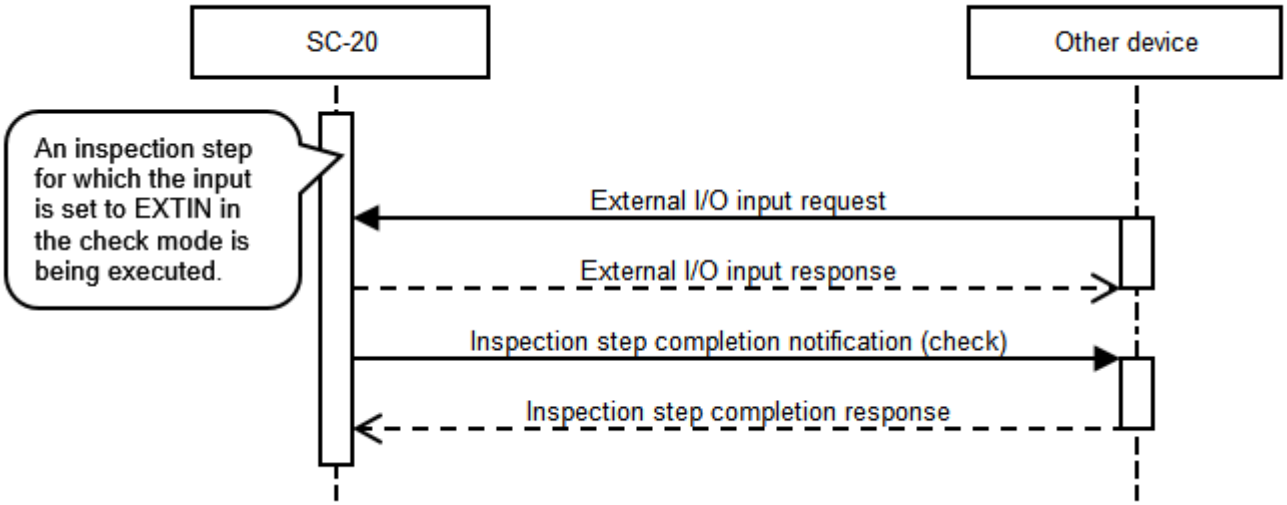

## Stopping an inspection step

To stop the processing during the “Inspection step being executed”, send a “Stop request” to the SC-20 to stop the job.



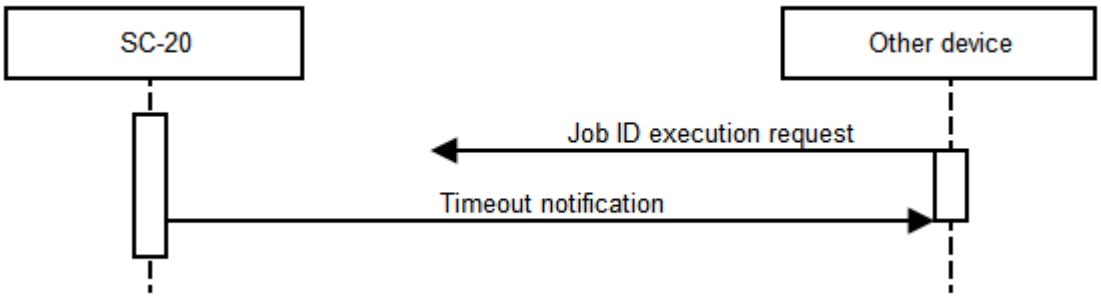
## External I/O input

If an “External I/O input request” is sent during the execution of an inspection step for which EXTIN is set in the check mode in advance, processing corresponding to the command parameters is executed, and an “External I/O input response” and “Inspection step completion notification” are sent from the SC-20 in a row.




## Timeout

If for some reason, the “Job ID execution request” from the other device does not reach the SC-20, a “Timeout notification” is sent to the other device after the transmission of a “Timeout notification” inside the system when the timer set inside the SC-20 expires. It is also likely that the “Timeout notification” does not reach due to a problem concerning the network path, and in such a case, check the network status at the other device side and take action.



# 3. Message ID

The message IDs used in the Socket Mode are described below.

## Socket Mode message IDs

### Requests

| Message name | | Message ID | |
|---|---|---|---|
| | | Other device → SC-20 | SC-20 → Other device |
| | | Request | Response |
| | Stop | 0x00000003 | 0x10000003 |
| | Acquiring inspection step list | 0x00000004 | 0x10000004 |
| | Job ID execution | 0x00000005 | 0x10000005 |
| | External I/O input | 0x00000007 | 0x10000007 |
| | Status check | 0x00000008 | 0x10000008 |
| | Shutdown execution | 0x00000009 | 0x10000009 |
| | Reboot execution | 0x0000000A | 0x1000000A |

### Notifications

| Message name | | Message ID | |
|---|---|---|---|
| | | SC-20 → Other device | Other device → SC-20 |
| | | Notification | Notification response |
| | Inspection step completion (matching) | 0x10010002 | 0x00010007 |
| | Inspection step completion (data input) | 0x10010003 | 0x00010007 |
| | Inspection step completion (check mode) | 0x10010004 | 0x00010007 |
| | Inspection step completion (stop) | 0x10010005 | 0x00010007 |
| | Job ID completion | 0x10010008 | 0x00010008 |
| | Inspection steps data | 0x10010009 | - |
| | Inspection step list acquisition completion | 0x1001000B | 0x0001000B |
| | Login | 0x1001000C | 0x0001000C |
| | Logout | 0x1001000D | 0x0001000D |
| | System outage | 0x1001000E | - |
| | Timeout | 0x1001000F | - |

# Message headers

A configuration chart of common message headers set in the data area of the Socket Mode is shown below.
As for the device ID and device name, use the data in the “Startup notification” sent during the startup of the SC-20.

**Message format**

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |

# Message IDs (Request messages)

## Stop request

| Message ID | Message name | Description |
|---|---|---|
| 0x00000003 | Stop request | If a “Stop request” is sent before an “Inspection step completion notification”, the processing being recognized is stopped. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |

## Stop response

| Message ID | Message name | Description |
|---|---|---|
| 0x10000003 | Stop response | A response message to a “Stop request” sent from the other device.<br>The result and error code are entered in the response message. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | Day | Month | Year | |
| 0x004c | reserve | Seconds | Minutes | Hours |
| 0x0050 | Error code | | Result | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 | 2 bytes | uint16 | Year | Set the time of the SC-20. |
| 0x004a | 1 byte | uchar | Month | Set the time of the SC-20 between 1 and 12. |
| 0x004b | 1 byte | uchar | Day | Set the time of the SC-20 between 1 and 31. |
| 0x004c | 1 byte | uchar | Hours | Set the time of the SC-20 between 0 and 23. |
| 0x004d | 1 byte | uchar | Minutes | Set the time of the SC-20 between 0 and 59. |
| 0x004e | 1 byte | uchar | Seconds | Set the time of the SC-20 between 0 and 59. |
| 0x004f | 1 byte | uchar | reserve | Unused area |
| 0x0050 | 2 bytes | int16 | Result | 0: OK<br>-1: FAIL |
| 0x0052 | 2 bytes | uint16 | Error code | See 4. Error code (P. 44). |

## Inspection step list acquisition request

| Message ID | Message name | Description |
|---|---|---|
| 0x00000004 | Inspection step list acquisition request | Acquire the inspection steps registered in the SC-20. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |

## Inspection step list acquisition response

| Message ID | Message name | Description |
|---|---|---|
| 0x10000004 | Inspection step list acquisition response | A response message to an “Inspection step list acquisition request” sent from the other device.<br>The number of inspection steps registered in the SC-20 is entered in the result (max. 32767).<br>In the case of a failure, -1 is returned, and a code indicating the reason is applied to the error code. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | **31** | **16** | **15** | **0** |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | Day | Month | Year | |
| 0x004c | reserve | Seconds | Minutes | Hours |
| 0x0050 | Error code | | Result | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 | 2 bytes | uint16 | Year | Set the time of the SC-20. |
| 0x004a | 1 byte | uchar | Month | Set the time of the SC-20 between 1 and 12. |
| 0x004b | 1 byte | uchar | Day | Set the time of the SC-20 between 1 and 31. |
| 0x004c | 1 byte | uchar | Hours | Set the time of the SC-20 between 0 and 23. |
| 0x004d | 1 byte | uchar | Minutes | Set the time of the SC-20 between 0 and 59. |
| 0x004e | 1 byte | uchar | Seconds | Set the time of the SC-20 between 0 and 59. |
| 0x004f | 1 byte | uchar | reserve | Unused area |
| 0x0050 | 2 bytes | int16 | Result | 1 to 32767: Total number of items<br>-1: FAIL |
| 0x0052 | 2 bytes | uint16 | Error code | See 4. Error code (P. 44). |

## Job ID execution request

| Message ID | Message name | Description |
|---|---|---|
| 0x00000005 | Job ID execution request | Start the execution of the job ID from the other device.<br>The state transits to ON if “RUN” is set in OUT0 of external I/O.<br>Note<br>• ON/OFF depends on user definition. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | Job ID | | | |
| 0x0088 | Instruction Step | | | |
| 0x00c8 | Inspection Step | | | |
| 0x0108 | User ID | | | |
| 0x0148 | Reference ID | | | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 - 0x0087 | 64 bytes | char | Job ID | Up to 50 single-byte alphanumeric characters. |
| 0x0088 - 0x00c7 | 64 bytes | char | Instruction Step | Up to 50 single-byte alphanumeric characters. |
| 0x00c8 - 0x0107 | 64 bytes | char | Inspection Step | Up to 50 single-byte alphanumeric characters. |
| 0x0108 - 0x0147 | 64 bytes | char | User ID | Up to 50 single-byte alphanumeric characters. |
| 0x0148 - 0x0187 | 64 bytes | char | Reference ID | Up to 50 single-byte alphanumeric characters. |

## Job ID execution response

| Message ID | Message name | Description |
|---|---|---|
| 0x10000005 | Job ID execution response | A response message to a “Job ID execution request” sent from the other device.<br>The result and error code are entered in the response message. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | Day | Month | Year | |
| 0x004c | reserve | Seconds | Minutes | Hours |
| 0x0050 | Error code | | Result | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 | 2 bytes | uint16 | Year | Set the time of the SC-20. |
| 0x004a | 1 byte | uchar | Month | Set the time of the SC-20 between 1 and 12. |
| 0x004b | 1 byte | uchar | Day | Set the time of the SC-20 between 1 and 31. |
| 0x004c | 1 byte | uchar | Hours | Set the time of the SC-20 between 0 and 23. |
| 0x004d | 1 byte | uchar | Minutes | Set the time of the SC-20 between 0 and 59. |
| 0x004e | 1 byte | uchar | Seconds | Set the time of the SC-20 between 0 and 59. |
| 0x004f | 1 byte | uchar | reserve | Unused area |
| 0x0050 | 2 bytes | int16 | Result | 0: OK<br>-1: FAIL |
| 0x0052 | 2 bytes | uint16 | Error code | See 4. Error code (P. 44). |

## External I/O input request

| Message ID | Message name | Description |
|---|---|---|
| 0x00000007 | External I/O input request | A message sent to the SC-20 when the inspection step is in standby in the check mode.<br>Perform the operation corresponding to EXTIN of external I/O. |

### Message format

| Address | bit | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 31 | | | | | | | 16 | 15 | | | | | | | 0 |
| 0x0000 | Message ID | | | | | | | | | | | | | | | |
| 0x0004 | Device ID | | | | | | | | | | | | | | | |
| 0x0008 | Device Name | | | | | | | | | | | | | | | |
| 0x0048 | Job ID | | | | | | | | | | | | | | | |
| 0x0088 | reserve | | | | | | | | | EXTIN9 | EXTIN8 | EXTIN7 | EXTIN6 | EXTIN5 | EXTIN4 | EXTIN3 | EXTIN2 | EXTIN1 | EXTIN0 |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 - 0x0087 | 64 bytes | char | Job ID | Up to 50 single-byte alphanumeric characters. |
| 0x0088 | 4 bytes | uint32 | External IO | Set the input logical value of external I/O in bit field.<br>[0]EXTIN0: 1/0<br>[1]EXTIN1: 1/0<br>[2]EXTIN2: 1/0<br>[3]EXTIN3: 1/0<br>[4]EXTIN4: 1/0<br>[5]EXTIN5: 1/0<br>[6]EXTIN6: 1/0<br>[7]EXTIN7: 1/0<br>[8]EXTIN8: 1/0<br>[9]EXTIN9: 1/0<br>[10-31]: reserve |

## External I/O input response

| Message ID | Message name | Description |
|---|---|---|
| 0x10000007 | External I/O input response | A response message to the “External I/O input request”.<br>The result and error code are entered in the response message. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | Day | Month | Year | |
| 0x004c | reserve | Seconds | Minutes | Hours |
| 0x0050 | Error code | | Result | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 | 2 bytes | uint16 | Year | Set the time of the SC-20. |
| 0x004a | 1 byte | uchar | Month | Set the time of the SC-20 between 1 and 12. |
| 0x004b | 1 byte | uchar | Day | Set the time of the SC-20 between 1 and 31. |
| 0x004c | 1 byte | uchar | Hours | Set the time of the SC-20 between 0 and 23. |
| 0x004d | 1 byte | uchar | Minutes | Set the time of the SC-20 between 0 and 59. |
| 0x004e | 1 byte | uchar | Seconds | Set the time of the SC-20 between 0 and 59. |
| 0x004f | 1 byte | uchar | reserve | Unused area |
| 0x0050 | 2 bytes | int16 | Result | 0: OK<br>-1: FAIL |
| 0x0052 | 2 bytes | uint16 | Error code | See 4. Error code (P. 44). |

## Status check request

| Message ID | Message name | Description |
|---|---|---|
| 0x00000008 | Status check request | A message for checking the current status of Socket Mode. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |

## Status check response

| Message ID | Message name | Description |
|---|---|---|
| 0x10000008 | Status check response | A response message to the “Status check request”.<br>The result and error code are entered in the response message.<br>It indicates the current camera status according to the result value. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | Day | Month | Year | |
| 0x004c | reserve | Seconds | Minutes | Hours |
| 0x0050 | Error code | | Result | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 | 2 bytes | uint16 | Year | Set the time of the SC-20. |
| 0x004a | 1 byte | uchar | Month | Set the time of the SC-20 between 1 and 12. |
| 0x004b | 1 byte | uchar | Day | Set the time of the SC-20 between 1 and 31. |
| 0x004c | 1 byte | uchar | Hours | Set the time of the SC-20 between 0 and 23. |
| 0x004d | 1 byte | uchar | Minutes | Set the time of the SC-20 between 0 and 59. |
| 0x004e | 1 byte | uchar | Seconds | Set the time of the SC-20 between 0 and 59. |
| 0x004f | 1 byte | uchar | reserve | Unused area |
| 0x0050 | 2 bytes | int16 | Result | Each state during Commands and System Status (→P. 8) is indicated by the following numbers:<br>2: Idle<br>3, 4: Inspection step being forwarded<br>5, 6: Job ID start<br>7, 13: Inspection step being executed<br>8, 9, 14: Job ID being executed<br>10, 11, 12: Job ID complete<br>15: Timeout<br>-1: FAIL |
| 0x0052 | 2 bytes | uint16 | Error code | See 4. Error code (P. 44). |

## Shutdown execution request

| Message ID | Message name | Description |
|---|---|---|
| 0x00000009 | Shutdown execution request | A message used when the SC-20 is to be shut down. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |

Note

- The shutdown execution request is not effective during log off.

## Shutdown execution response

| Message ID | Message name | Description |
|---|---|---|
| 0x10000009 | Shutdown execution response | A response message to a “Shutdown execution request”.<br>The result and error code are entered in the response message. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | Day | Month | Year | |
| 0x004c | reserve | Seconds | Minutes | Hours |
| 0x0050 | Error code | | Result | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 | 2 bytes | uint16 | Year | Set the time of the SC-20. |
| 0x004a | 1 byte | uchar | Month | Set the time of the SC-20 between 1 and 12. |
| 0x004b | 1 byte | uchar | Day | Set the time of the SC-20 between 1 and 31. |
| 0x004c | 1 byte | uchar | Hours | Set the time of the SC-20 between 0 and 23. |
| 0x004d | 1 byte | uchar | Minutes | Set the time of the SC-20 between 0 and 59. |
| 0x004e | 1 byte | uchar | Seconds | Set the time of the SC-20 between 0 and 59. |
| 0x004f | 1 byte | uchar | reserve | Unused area |
| 0x0050 | 2 bytes | int16 | Result | 0: OK<br>-1: FAIL |
| 0x0052 | 2 bytes | uint16 | Error code | See 4. Error code (P. 44). |

## Reboot execution request

| Message ID | Message name | Description |
|---|---|---|
| 0x0000000A | Reboot execution request | A message used when the SC-20 is to be rebooted. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |

Note

- The reboot execution request is not effective during log off.

## Reboot execution response

| Message ID | Message name | Description |
|---|---|---|
| 0x1000000A | Reboot execution response | A response message to a “Reboot execution request”.<br>The result and error code are entered in the response message. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | Day | Month | Year | |
| 0x004c | reserve | Seconds | Minutes | Hours |
| 0x0050 | Error code | | Result | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 | 2 bytes | uint16 | Year | Set the time of the SC-20. |
| 0x004a | 1 byte | uchar | Month | Set the time of the SC-20 between 1 and 12. |
| 0x004b | 1 byte | uchar | Day | Set the time of the SC-20 between 1 and 31. |
| 0x004c | 1 byte | uchar | Hours | Set the time of the SC-20 between 0 and 23. |
| 0x004d | 1 byte | uchar | Minutes | Set the time of the SC-20 between 0 and 59. |
| 0x004e | 1 byte | uchar | Seconds | Set the time of the SC-20 between 0 and 59. |
| 0x004f | 1 byte | uchar | reserve | Unused area |
| 0x0050 | 2 bytes | int16 | Result | 0: OK<br>-1: FAIL |
| 0x0052 | 2 bytes | uint16 | Error code | See 4. Error code (P. 44). |

# Message IDs (Notification messages)

## Inspection step completion notification (matching)

| Message ID | Message name | Description |
|---|---|---|
| 0x10010002 | Inspection step completion notification (matching) | A message sent to the other device when the matching processing ends.<br>The data in “Checkpoint ID_X” and later is input according to the number set in “Number of check points”. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | Day | Month | Year | |
| 0x004c | reserve | Seconds | Minutes | Hours |
| 0x0050 | Job ID | | | |
| 0x0090 | Instruction Step | | | |
| 0x00d0 | Inspection Step | | | |
| 0x0110 | User ID | | | |
| 0x01d8 | Reference ID | | | |
| 0x02a0 | Elapsed time | | Inspection step final result | |
| 0x02a4 | Anchor point similarity | | | |
| 0x02ac | Number of check points | | Anchor point rotation angle | |
| 0x02b0 | reserve | Judgment result | Mode (matching) | Checkpoint ID_1 |
| 0x02b4 | Matching time [msec] | | Rotation Angle | |
| 0x02b8 | Similarity | | | |
| 0x02c0 | reserve | Judgment result | Mode (matching) | Checkpoint ID_2 |
| 0x02c4 | Matching time [msec] | | Rotation Angle | |
| 0x02c8 | Similarity | | | |
| ~ | ~ | | | |
| 0x04a0 | reserve | Judgment result | Mode (matching) | Checkpoint ID_20 |
| 0x04a4 | Matching time [msec] | | Rotation Angle | |
| 0x04a8 | Similarity | | | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 | 2 bytes | uint16 | Year | Set the time of the SC-20. |
| 0x004a | 1 byte | uchar | Month | Set the time of the SC-20 between 1 and 12. |
| 0x004b | 1 byte | uchar | Day | Set the time of the SC-20 between 1 and 31. |
| 0x004c | 1 byte | uchar | Hours | Set the time of the SC-20 between 0 and 23. |
| 0x004d | 1 byte | uchar | Minutes | Set the time of the SC-20 between 0 and 59. |
| 0x004e | 1 byte | uchar | Seconds | Set the time of the SC-20 between 0 and 59. |
| 0x004f | 1 byte | uchar | reserve | Unused area |
| 0x0050 - 0x008f | 64 bytes | char | Job ID | Up to 50 single-byte alphanumeric characters. |
| 0x0090 - 0x00cf | 64 bytes | char | Instruction Step | Up to 50 single-byte alphanumeric characters. |
| 0x00d0 - 0x010f | 64 bytes | char | Inspection Step | Up to 50 single-byte alphanumeric characters. |
| 0x0110 - 0x01d7 | 200 bytes | char | User ID | Up to 198 single-byte alphanumeric characters. |
| 0x01d8 - 0x029f | 200 bytes | char | Reference ID | Up to 198 single-byte alphanumeric characters. |
| 0x02a0 | 2 bytes | int16 | Inspection step final result | 0: OK<br>-1: FAIL<br>-2: Anchor point failure |
| 0x02a2 | 2 bytes | uint16 | Elapsed time (sec) | Set the job elapsed time in seconds. |
| 0x02a4 | 8 bytes | double | Anchor Point Similarity | Set from 0.00000 to 1.00000. |
| 0x02ac | 2 bytes | int16 | Anchor Point Rotation Angle | Set from 180 to -180. |
| 0x02ae | 2 bytes | uint16 | Number of check points | Set the number of check points from 0 to 20. |
| 0x02b0 | 1 byte | uchar | Checkpoint ID_1 | Set the checkpoint ID from 1 to 20. |
| 0x02b1 | 1 byte | uchar | Mode | 0: Shape<br>1: Color Recognition<br>2: Texture |
| 0x02b2 | 1 byte | char | Judgment result | 0: OK<br>-1: FAIL |
| 0x02b3 | 1 byte | uchar | reserve | Unused area |
| 0x02b4 | 2 bytes | int16 | Rotation Angle | Set from 180 to -180.<br>* Color Recognition and Texture are fixed as 0. |
| 0x02b6 | 2 bytes | uint16 | Matching time [msec] | Set from 0 to 65535. |
| 0x02b8 | 8 bytes | double | Similarity | Set from 0.00000 to 1.00000. |

## Inspection step completion notification (data input)

| Message ID | Message name | Description |
|---|---|---|
| 0x10010003 | Inspection step completion notification (data input) | A message sent to the other device when data input ends. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | Day | Month | Year | |
| 0x004c | reserve | Seconds | Minutes | Hours |
| 0x0050 | Job ID | | | |
| 0x0090 | Instruction Step | | | |
| 0x00d0 | Inspection Step | | | |
| 0x0110 | User ID | | | |
| 0x01d8 | Reference ID | | | |
| 0x02a0 | Elapsed time (sec) | | Inspection step final result | |
| 0x02a4 | Part No. (an already set number) | | | |
| 0x0324 | Input data (a value entered by the user) | | | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 | 2 bytes | uint16 | Year | Set the time of the SC-20. |
| 0x004a | 1 byte | uchar | Month | Set the time of the SC-20 between 1 and 12. |
| 0x004b | 1 byte | uchar | Day | Set the time of the SC-20 between 1 and 31. |
| 0x004c | 1 byte | uchar | Hours | Set the time of the SC-20 between 0 and 23. |
| 0x004d | 1 byte | uchar | Minutes | Set the time of the SC-20 between 0 and 59. |
| 0x004e | 1 byte | uchar | Seconds | Set the time of the SC-20 between 0 and 59. |
| 0x004f | 1 byte | uchar | reserve | Unused area |
| 0x0050 - 0x008f | 64 bytes | char | Job ID | Up to 50 single-byte alphanumeric characters. |
| 0x0090 - 0x00cf | 64 bytes | char | Instruction Step | Up to 50 single-byte alphanumeric characters. |
| 0x00d0 - 0x010f | 64 bytes | char | Inspection Step | Up to 50 single-byte alphanumeric characters. |
| 0x0110 - 0x01d7 | 200 bytes | char | User ID | Up to 198 single-byte alphanumeric characters. |
| 0x01d8 - 0x029f | 200 bytes | char | Reference ID | Up to 198 single-byte alphanumeric characters. |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x02a0 | 2 bytes | int16 | Inspection step final result | 0: OK<br>-1: FAIL |
| 0x02a2 | 2 bytes | uint16 | Elapsed time (sec) | Set the job elapsed time in seconds. |
| 0x02a4 - 0x0323 | 128 bytes | char | Part No. | Single-byte alphanumeric characters |
| 0x0324 - 0x0523 | 512 bytes | char | Input data | Single-byte alphanumeric characters |

## Inspection step completion notification (check mode)

| Message ID | Message name | Description |
|---|---|---|
| 0x10010004 | Inspection step completion notification (check mode) | A message sent to the other device when the check mode ends. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | Day | Month | Year | |
| 0x004c | reserve | Seconds | Minutes | Hours |
| 0x0050 | Job ID | | | |
| 0x0090 | Instruction Step | | | |
| 0x00d0 | Inspection Step | | | |
| 0x0110 | User ID | | | |
| 0x01d8 | Reference ID | | | |
| 0x02a0 | Elapsed time (sec) | | Inspection step final result | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 | 2 bytes | uint16 | Year | Set the time of the SC-20. |
| 0x004a | 1 byte | uchar | Month | Set the time of the SC-20 between 1 and 12. |
| 0x004b | 1 byte | uchar | Day | Set the time of the SC-20 between 1 and 31. |
| 0x004c | 1 byte | uchar | Hours | Set the time of the SC-20 between 0 and 23. |
| 0x004d | 1 byte | uchar | Minutes | Set the time of the SC-20 between 0 and 59. |
| 0x004e | 1 byte | uchar | Seconds | Set the time of the SC-20 between 0 and 59. |
| 0x004f | 1 byte | uchar | reserve | Unused area |
| 0x0050 - 0x008f | 64 bytes | char | Job ID | Up to 50 single-byte alphanumeric characters. |
| 0x0090 - 0x00cf | 64 bytes | char | Instruction Step | Up to 50 single-byte alphanumeric characters. |
| 0x00d0 - 0x010f | 64 bytes | char | Inspection Step | Up to 50 single-byte alphanumeric characters. |
| 0x0110 - 0x01d7 | 200 bytes | char | User ID | Up to 198 single-byte alphanumeric characters. |
| 0x01d8 - 0x029f | 200 bytes | char | Reference ID | Up to 198 single-byte alphanumeric characters. |
| 0x02a0 | 2 bytes | int16 | Inspection step final result | 0: OK<br>-1: FAIL |
| 0x02a2 | 2 bytes | uint16 | Elapsed time (sec) | Set the job elapsed time in seconds. |

## Inspection step completion notification (stop)

| Message ID | Message name | Description |
|---|---|---|
| 0x10010005 | Inspection step completion notification (stop) | A message sent from the SC-20 after the execution of Stop from the UI, external I/O, and Socket Mode. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | Day | Month | Year | |
| 0x004c | reserve | Seconds | Minutes | Hours |
| 0x0050 | Job ID | | | |
| 0x0090 | Instruction Step | | | |
| 0x00d0 | Inspection Step | | | |
| 0x0110 | Elapsed time (sec) | | Stop factor | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 | 2 bytes | uint16 | Year | Set the time of the SC-20. |
| 0x004a | 1 byte | uchar | Month | Set the time of the SC-20 between 1 and 12. |
| 0x004b | 1 byte | uchar | Day | Set the time of the SC-20 between 1 and 31. |
| 0x004c | 1 byte | uchar | Hours | Set the time of the SC-20 between 0 and 23. |
| 0x004d | 1 byte | uchar | Minutes | Set the time of the SC-20 between 0 and 59. |
| 0x004e | 1 byte | uchar | Seconds | Set the time of the SC-20 between 0 and 59. |
| 0x004f | 1 byte | uchar | reserve | Unused area |
| 0x0050 - 0x008f | 64 bytes | char | Job ID | Up to 50 single-byte alphanumeric characters. |
| 0x0090 - 0x00cf | 64 bytes | char | Instruction Step | Up to 50 single-byte alphanumeric characters. |
| 0x00d0 - 0x010f | 64 bytes | char | Inspection Step | Up to 50 single-byte alphanumeric characters. |
| 0x0110 | 2 bytes | int16 | Stop factor | 0: Stopped from the UI<br>1: Stopped from the external I/O<br>2: Stopped from the Socket Mode |
| 0x0102 | 2 bytes | uint16 | Elapsed time (sec) | Set the job elapsed time in seconds. |

## Inspection step completion notification response

| Message ID | Message name | Description |
|---|---|---|
| 0x00010007 | Inspection step completion notification response | A response message to an “Inspection step completion notification” sent from the SC-20. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | reserve | | | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 | 4 bytes | Uint32 | reserve | Unused area |

## Job ID completion notification

| Message ID | Message name | Description |
|---|---|---|
| 0x10010008 | Job ID completion notification | A message sent from the SC-20 after the execution of all inspection steps managed in the job ID is complete.<br>Send a “Job ID completion notification” even when “2” is set in the “Result” field of the “Inspection step completion notification”. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | Day | Month | Year | |
| 0x004c | reserve | Seconds | Minutes | Hours |
| 0x0050 | Job ID | | | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 | 2 bytes | uint16 | Year | Set the time of the SC-20. |
| 0x004a | 1 byte | uchar | Month | Set the time of the SC-20 between 1 and 12. |
| 0x004b | 1 byte | uchar | Day | Set the time of the SC-20 between 1 and 31. |
| 0x004c | 1 byte | uchar | Hours | Set the time of the SC-20 between 0 and 23. |
| 0x004d | 1 byte | uchar | Minutes | Set the time of the SC-20 between 0 and 59. |
| 0x004e | 1 byte | uchar | Seconds | Set the time of the SC-20 between 0 and 59. |
| 0x004f | 1 byte | uchar | reserve | Unused area |
| 0x0050 - 0x008f | 64 bytes | char | Job ID | Up to 50 single-byte alphanumeric characters. |

## Job ID completion notification response

| Message ID | Message name | Description |
|---|---|---|
| 0x00010008 | Job ID completion notification response | A response message to a “Job ID completion notification” sent from the SC-20.<br>A series of operations are completed when the “Job ID completion notification response” is received. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |

## Inspection steps data notification

| Message ID | Message name | Description |
|---|---|---|
| 0x10010009 | Inspection steps data notification | A message notifying the job ID, instruction step, and inspection step information in response to the “Inspection step list acquisition request”. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | Day | Month | Year | |
| 0x004c | reserve | Seconds | Minutes | Hours |
| 0x0050 | Job ID | | | |
| 0x0090 | Instruction Step | | | |
| 0x00d0 | Inspection Step | | | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 | 2 bytes | uint16 | Year | Set the time of the SC-20. |
| 0x004a | 1 byte | uchar | Month | Set the time of the SC-20 between 1 and 12. |
| 0x004b | 1 byte | uchar | Day | Set the time of the SC-20 between 1 and 31. |
| 0x004c | 1 byte | uchar | Hours | Set the time of the SC-20 between 0 and 23. |
| 0x004d | 1 byte | uchar | Minutes | Set the time of the SC-20 between 0 and 59. |
| 0x004e | 1 byte | uchar | Seconds | Set the time of the SC-20 between 0 and 59. |
| 0x004f | 1 byte | uchar | reserve | Unused area |
| 0x0050 - 0x008f | 64 bytes | char | Job ID | Up to 50 single-byte alphanumeric characters. |
| 0x0090 - 0x00cf | 64 bytes | char | Instruction Step | Up to 50 single-byte alphanumeric characters. |
| 0x00d0 - 0x010f | 64 bytes | char | Inspection Step | Up to 50 single-byte alphanumeric characters. |

## Inspection step list acquisition completion notification

| Message ID | Message name | Description |
|---|---|---|
| 0x1001000B | Inspection step list acquisition completion notification | A message sent to the other device after the completion of transmission of all “Inspection steps data notification”.<br>Enter the number of inspection steps sent to the “Number of transfers” field. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | Day | Month | Year | |
| 0x004c | reserve | Seconds | Minutes | Hours |
| 0x0050 | Error code | | Number of transfers | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 | 2 bytes | uint16 | Year | Set the time of the SC-20. |
| 0x004a | 1 byte | uchar | Month | Set the time of the SC-20 between 1 and 12. |
| 0x004b | 1 byte | uchar | Day | Set the time of the SC-20 between 1 and 31. |
| 0x004c | 1 byte | uchar | Hours | Set the time of the SC-20 between 0 and 23. |
| 0x004d | 1 byte | uchar | Minutes | Set the time of the SC-20 between 0 and 59. |
| 0x004e | 1 byte | uchar | Seconds | Set the time of the SC-20 between 0 and 59. |
| 0x004f | 1 byte | uchar | reserve | Unused area |
| 0x0050 | 2 bytes | int16 | Number of transfers | 1 to 32767: Total number of items<br>-1: FAIL |
| 0x0052 | 2 bytes | uint16 | Error code | See 4. Error code (P. 44). |

## Inspection step list acquisition completion notification response

| Message ID | Message name | Description |
|---|---|---|
| 0x0001000B | Inspection step list acquisition completion notification response | A response message to an “Inspection step list acquisition completion notification” sent from the SC-20.<br>A series of operations are completed through the “Inspection step list acquisition completion notification response”. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |

## System outage notification

| Message ID | Message name | Description |
|---|---|---|
| 0x1001000E | System outage notification | A message sent to the other device after shutting down or restarting the SC-20.<br>No response is sent to this message. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | Day | Month | Year | |
| 0x004c | reserve | Seconds | Minutes | Hours |
| 0x0050 | Stop mode | | | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 | 2 bytes | uint16 | Year | Set the time of the SC-20. |
| 0x004a | 1 byte | uchar | Month | Set the time of the SC-20 between 1 and 12. |
| 0x004b | 1 byte | uchar | Day | Set the time of the SC-20 between 1 and 31. |
| 0x004c | 1 byte | uchar | Hours | Set the time of the SC-20 between 0 and 23. |
| 0x004d | 1 byte | uchar | Minutes | Set the time of the SC-20 between 0 and 59. |
| 0x004e | 1 byte | uchar | Seconds | Set the time of the SC-20 between 0 and 59. |
| 0x004f | 1 byte | uchar | reserve | Unused area |
| 0x0050 | 4 bytes | uint32 | Stop mode | 0: Shutdown<br>1: Reboot |

## Timeout notification

| Message ID | Message name | Description |
|---|---|---|
| 0x1001000F | Timeout notification | A message sent to the other device when messages cannot be processed in the SC-20 due to some reason.<br>No response is sent to this message. |

### Message format

| Address | bit | | | |
|---|---|---|---|---|
| | 31 | 16 | 15 | 0 |
| 0x0000 | Message ID | | | |
| 0x0004 | Device ID | | | |
| 0x0008 | Device Name | | | |
| 0x0048 | Day | Month | Year | |
| 0x004c | reserve | Seconds | Minutes | Hours |
| 0x0050 | Error code | | Result | |

| Address | Size | Attribute | Field name | Description |
|---|---|---|---|---|
| 0x0000 | 4 bytes | uint32 | Message ID | An ID unique to the message. |
| 0x0004 | 4 bytes | uint32 | Device ID | An ID unique to the device. |
| 0x0008 - 0x0047 | 64 bytes | char | Device Name | Up to 50 single-byte alphanumeric characters. |
| 0x0048 | 2 bytes | uint16 | Year | Set the time of the SC-20. |
| 0x004a | 1 byte | uchar | Month | Set the time of the SC-20 between 1 and 12. |
| 0x004b | 1 byte | uchar | Day | Set the time of the SC-20 between 1 and 31. |
| 0x004c | 1 byte | uchar | Hours | Set the time of the SC-20 between 0 and 23. |
| 0x004d | 1 byte | uchar | Minutes | Set the time of the SC-20 between 0 and 59. |
| 0x004e | 1 byte | uchar | Seconds | Set the time of the SC-20 between 0 and 59. |
| 0x004f | 1 byte | uchar | reserve | Unused area |
| 0x0050 | 2 bytes | int16 | Result | 0: OK<br>-1: FAIL |
| 0x0052 | 2 bytes | uint16 | Error code | See 4. Error code (P. 44). |

# 4. Error code

| Error code | Error name | Cause | Target message | Action to take |
|---|---|---|---|---|
| 1 | Unknown device ID | The device ID sent to the SC-20 and the set device ID are mismatching. | Request messages in general | Make sure the device ID and device name in the sent message match those in the SC-20.<br>Make sure there is no mistake with upper case characters and lower case characters in the device name. |
| 2 | Unknown device name | The device name sent to the SC-20 and the set device name are mismatching. | Request messages in general | |
| 101 | Status transition failure | A job ID start request is received in other than the standby state. | Job ID start request | The “Job ID start request” and “Job ID execution request” can be received only after login or after the completion of the job ID.<br>The request could have been sent either during matching or before the “Job ID completion response”. |
| 102 | | A job ID execution request is received in other than the standby state. | Job ID execution request | |
| 103 | Status transition failure | A start request is received in other than the ready state. | Start request | The start request can be received only after the “Job ID start request” or after the “Inspection step completion response”.<br>It cannot be received during the execution of a job ID. |
| 104 | Status transition failure | A stop request is received in other than the run state. | Stop request | The stop request can only be received during matching. This error code is displayed even when matching is complete before a stop request. |
| 105 | Status transition failure | An inspection step list acquisition request is received in other than the standby state. | Inspection step list acquisition request | The inspection step list acquisition request can be received only after login or after completion of the job ID.<br>The request could have been sent either during matching or before the “Job ID completion response”. |
| 106 | Status transition failure | An inspection step list acquisition request is received in the user mode. | Inspection step list acquisition request | The inspection step list acquisition request can only be made in the administrator mode. Log in with the administrator mode. |
| 107 | Status transition failure | A job ID change request is received in other than the standby state. | Job ID change request | The inspection step list acquisition request can be received only after login or after completion of the job ID. |

| Error code | Error name | Cause | Target message | Action to take |
|---|---|---|---|---|
| | | | | The request could have been sent either during matching or before the “Job ID completion response”. |
| 108 | EXTIN input | An EXTIN input request is received other than during matching. | EXTIN input request | The EXTIN input request can only be received during matching. |
| 109 | Logging out | A request message is received when the user is logging out. | Request messages in general | Perform login. |
| 201 | Job ID name mismatch | The specified job ID does not exist. | Start request<br>Job ID execution request | The job ID, instruction steps, and inspection step entered in the request did not match.<br>Make sure the instruction step name was not changed in the SC-20.<br>Make sure no unnecessary space is entered at the beginning or end. |
| 202 | Instruction Step List name mismatch | The specified instruction step does not exist. | Start request<br>Job ID execution request | |
| 203 | Inspection step name mismatch | The specified inspection step does not exist. | Start request<br>Job ID execution request | |
| 204 | Job ID name blank | The job ID name is not specified. | Start request<br>Job ID execution request | Enter the job ID name in the specified format. |
| 207 | Busy status | The camera is in the processing state and a request to start the job ID has failed. | Job ID start request | Request again after some time. |
| 208 | Busy status | The camera is in the processing state and a request to change the job ID has failed. | Job ID change request | Request again after some time. |
| 209 | Busy status | The camera times out on consecutive requests. | Job ID start request, Job ID execution request | Request again after some time. |
| 210 | Extin Input | The entered I/O is invalid, or matching has stopped. | Extin input request | Enter a value within the normal range on the matching screen. |
| 301 | Matching result generation failure | Failed to generate the matching result. | Inspection step completion notification | Displayed in the case of failure in generating the matching data. |
| 401 | Timeout | A response cannot be received to the notification. | Timeout notification | Displayed when there is no response for 3 seconds after sending a “Job ID start response”, an “Inspection step completion notification”, or a “Job ID completion notification”. |
| 550 | Connection error NomachIPError | Data is received from an unknown IP address. | Messages in general | Send from the set IP address. |

# 5. Sample Code

The sample code in C language is described below. This code describes the sample of one path of job ID execution (P. 12 Job ID execution processing) under the assumption that the job ID “Default”, list “Work_1”, and item “Item_1” are already registered in the SC-20 unit as the matching mode.
Moreover, this code is based on Windows and uses winsock2. Therefore, you will need to link to the “ws2_32.lib” library. This library can be acquired by installing Visual Studio or the like.

```
#include <stdio.h>
#include <winsock2.h>       // need ws2_32.lib
#include <ws2tcpip.h>

// char* -> short
short char2short(char* c)
{
    short ret = 0;
    ret |= ((short)c[0] & 0x00FF);
    ret |= (((short)c[1] << 8) & 0xFF00);
    return ret;
}

// char* -> unsigned short
unsigned short char2ushort(char* c)
{
    unsigned short ret = 0;
    ret |= ((unsigned short)c[0] & 0x00FF);
    ret |= (((unsigned short)c[1] << 8) & 0xFF00);
    return ret;
}

// char* -> double
double char2double(char* c)
{
    double ret = 0.0;
    memcpy(&ret, c, 8);
    return ret;
}

// main
int main()
{
    int ret = 0;
    unsigned short port = 56109; // IP Port Number (Default=56109)
    unsigned int termID = 2030446878; // Terminal ID (unique)
    char termName[]= "SC20"; // Terminal Name (Default="SC20")
    SOCKET srcSocket;  // My PC Socket
    SOCKET dstSocket;  // SC20 Socket

    // sockaddr_in
    struct sockaddr_in srcAddr;
    struct sockaddr_in dstAddr;
    int dstAddrSize = sizeof(dstAddr);

    // buffer
    char rcvBuffer[4096];
    char tmpBuffer[128];
```

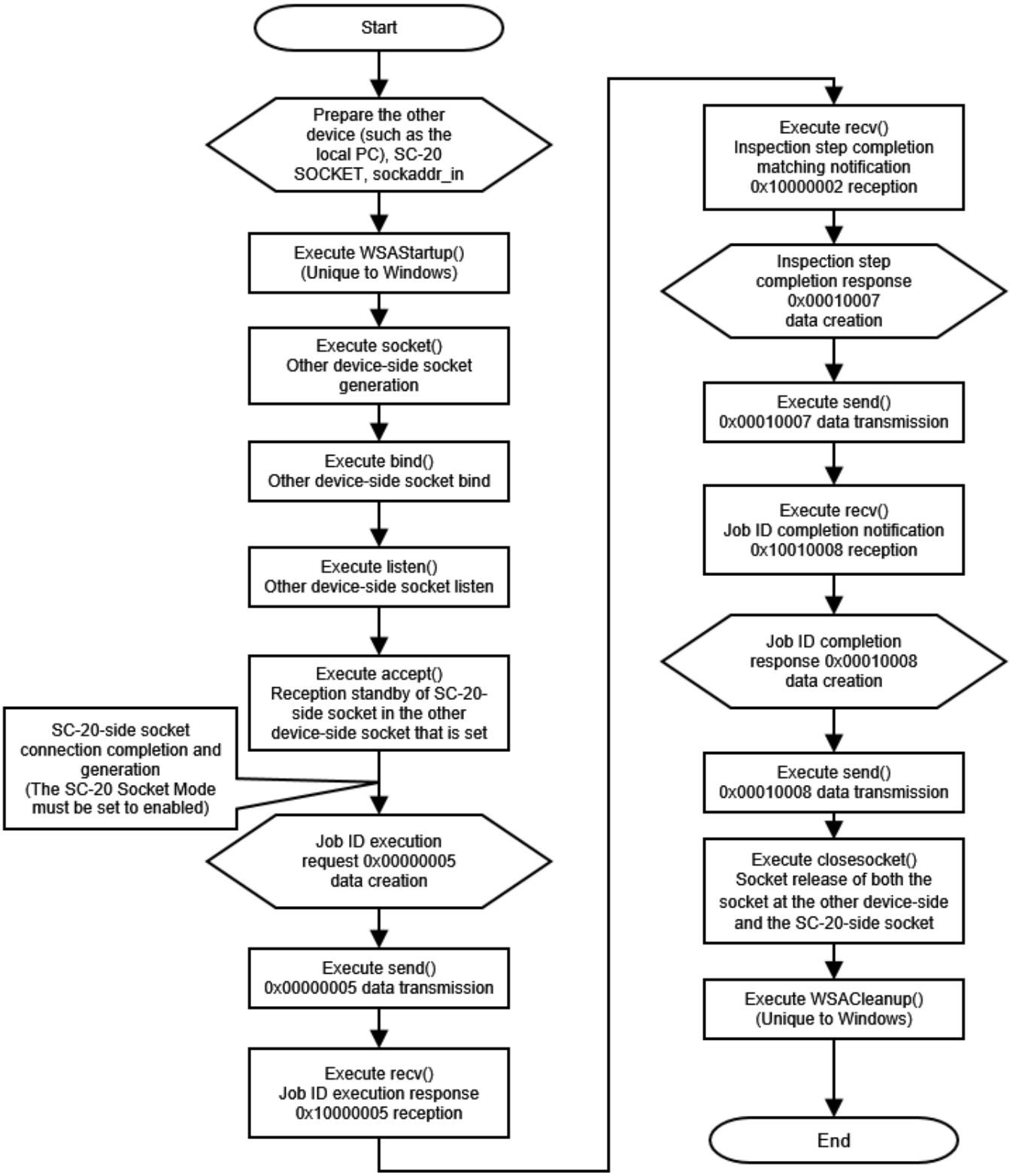
```
char sndBuffer1[0x0188];
char sndBuffer2[0x004c];

// Windows Only Process
WSADATA data;
WSAStartup(MAKEWORD(2,0), &data);

// sockaddr_in Setting
memset(&srcAddr, 0, sizeof(srcAddr));
srcAddr.sin_port = htons(port);
srcAddr.sin_family = AF_INET;
srcAddr.sin_addr.s_addr = htonl(INADDR_ANY);

// My PC Socket Create
srcSocket = socket(AF_INET, SOCK_STREAM, 0);

// My PC Socket Bind
bind(srcSocket, (struct sockaddr *) &srcAddr, sizeof(srcAddr));

// Connect Listen
listen(srcSocket, 1);

// Connect Accept <- SC20
printf("Waiting for connection ...¥n");
dstSocket = accept(srcSocket, (struct sockaddr *) &dstAddr, &dstAddrSize);
printf("Connected from %s¥n¥n", inet_ntoa(dstAddr.sin_addr));

////////////////////////////////////////
// Work ID Execute Command
////////////////////////////////////////

// 0x00000005 Request Event Buffer
memset(sndBuffer1, 0, 0x0188);
sndBuffer1[0x0000] = 0x05;
sndBuffer1[0x0001] = 0x00;
sndBuffer1[0x0002] = 0x00;
sndBuffer1[0x0003] = 0x00;
sndBuffer1[0x0004] = (char)((termID & 0x000000FF) >>  0);
sndBuffer1[0x0005] = (char)((termID & 0x0000FF00) >>  8);
sndBuffer1[0x0006] = (char)((termID & 0x00FF0000) >> 16);
sndBuffer1[0x0007] = (char)((termID & 0xFF000000) >> 24);
sprintf(&sndBuffer1[0x0008], termName);
sprintf(&sndBuffer1[0x0048], "Default");
sprintf(&sndBuffer1[0x0088], "Work_1");
sprintf(&sndBuffer1[0x00C8], "Item_1");
sprintf(&sndBuffer1[0x0108], "User");
sprintf(&sndBuffer1[0x0148], "1234567890");

// 0x00000005 Request Send -> SC-20
ret = send(dstSocket, sndBuffer1, 0x0188, 0);
if(0 > ret){
    printf("Socket Send Error¥n");
    return -1;
}


// 0x10000005 Response Recieve <- SC-20
```

```
    ret = recv(dstSocket, rcvBuffer, 4096, 0);
    if(0 >= ret){
        printf("Socket Recieve Error¥n");
        return -1;
    }

    // Response  Buffur Output
    printf("¥n-------------------------------------------------------------¥n");
    printf("@ Work ID Execute Command Response¥n");
    printf("-------------------------------------------------------------¥n¥n");
    printf("MsgID     : 0x%02X%02X%02X%02X¥n", rcvBuffer[3], rcvBuffer[2], rcvBuffer[1], rcvBuffer[0]);
    printf("TermID    : %d¥n", ((unsigned int)rcvBuffer[7] << 24) | ((unsigned int)rcvBuffer[6] << 16) | ((unsigned
int)rcvBuffer[5] << 8) | ((unsigned int)rcvBuffer[4]));    strcpy(tmpBuffer, &rcvBuffer[8]);
    printf("TermName  : %s¥n", tmpBuffer);
    printf("Date      : %04d/%02d/%02d %02d:%02d:%02d¥n", char2short(&rcvBuffer[0x48])
                                                          , rcvBuffer[0x4A]
                                                          , rcvBuffer[0x4B]
                                                          , rcvBuffer[0x4C]
                                                          , rcvBuffer[0x4D]
                                                          , rcvBuffer[0x4E]
    );
    short result = ((short)rcvBuffer[0x51] << 8) | (short)rcvBuffer[0x50];
    printf("Result    : %d¥n", result);
    printf("ErrorCode : %d¥n", (((unsigned short)rcvBuffer[0x53] & 0xFF) << 8) | ((unsigned short)rcvBuffer[0x52] &
0xFF));
    if(0 > result){
        printf("0x00000005 Command Error¥n");
        return -1;
    }

    ////////////////////////////////////////
    // Work Item Done (Matching) Notify
    ////////////////////////////////////////

    // 0x10010002 Notify Recieve <- SC-20
    ret = recv(dstSocket, rcvBuffer, 4096, 0);
    if(0 >= ret){
        printf("Socket Recieve Error¥n");
        return -1;
    }

    // Work Item Done (Matching) Buffer Output
    printf("¥n-------------------------------------------------------------¥n");
    printf("@ Work Item Done (Matching) Notify¥n");
    printf("-------------------------------------------------------------¥n¥n");
    printf("MsgID     : 0x%02X%02X%02X%02X¥n", rcvBuffer[3], rcvBuffer[2], rcvBuffer[1], rcvBuffer[0]);
    printf("TermID    : %d¥n", ((unsigned int)rcvBuffer[7] << 24) | ((unsigned int)rcvBuffer[6] << 16) | ((unsigned
int)rcvBuffer[5] << 8) | ((unsigned int)rcvBuffer[4]));
    strcpy(tmpBuffer, &rcvBuffer[8]);
    printf("TermName  : %s¥n", tmpBuffer);
    printf("Date      : %04d/%02d/%02d %02d:%02d:%02d¥n", char2short(&rcvBuffer[0x48])
                                                        , rcvBuffer[0x4A]
                                                        , rcvBuffer[0x4B]
                                                        , rcvBuffer[0x4C]
                                                        , rcvBuffer[0x4D]
                                                        , rcvBuffer[0x4E]
    );
```

```
strcpy(tmpBuffer, &rcvBuffer[0x0050]);
printf("WorkID     : %s¥n", tmpBuffer);
strcpy(tmpBuffer, &rcvBuffer[0x0090]);
printf("WorkList   : %s¥n", tmpBuffer);
strcpy(tmpBuffer, &rcvBuffer[0x00D0]);
printf("WorkItem   : %s¥n", tmpBuffer);
strcpy(tmpBuffer, &rcvBuffer[0x0110]);
printf("WorkerID   : %s¥n", tmpBuffer);
strcpy(tmpBuffer, &rcvBuffer[0x01D8]);
printf("WorkNum    : %s¥n", tmpBuffer);
printf("Result     : %d¥n", char2short(&rcvBuffer[0x02A0]));
printf("Time       : %d¥n", char2ushort(&rcvBuffer[0x02A2]));
printf("BaseScore : %f¥n", char2double(&rcvBuffer[0x02A4]));
printf("BaseAngle : %d¥n", char2short(&rcvBuffer[0x02AC]));
int checkNum = (int)char2short(&rcvBuffer[0x02AE]);
printf("CheckNum   : %d¥n", checkNum);
for(int i = 0; i < checkNum; i++){
    int offset = i * 16;
    printf("CheckIndex[%d]¥n", rcvBuffer[0x02B0 + offset]);

    printf("  Mode     : %d¥n", rcvBuffer[0x02B1 + offset]);
    printf("  Result   : %d¥n", char2short(&rcvBuffer[0x02B2]));
    printf("  Angle    : %d¥n", char2short(&rcvBuffer[0x02B4]));
    printf("  Time     : %d¥n", char2ushort(&rcvBuffer[0x02B6]));
    printf("  Score    : %f¥n", char2double(&rcvBuffer[0x02B8]));
}

// 0x00010007 Work Item Done (Matching) Notify Response Send -> SC-20
memset(sndBuffer2, 0, 0x004c);
sndBuffer2[0x0000] = 0x07;
sndBuffer2[0x0001] = 0x00;
sndBuffer2[0x0002] = 0x01;
sndBuffer2[0x0003] = 0x00;
sndBuffer2[0x0004] = (char)((termID & 0x000000FF) >>  0);
sndBuffer2[0x0005] = (char)((termID & 0x0000FF00) >>  8);
sndBuffer2[0x0006] = (char)((termID & 0x00FF0000) >> 16);
sndBuffer2[0x0007] = (char)((termID & 0xFF000000) >> 24);
sprintf(&sndBuffer2[0x0008], termName);
ret = send(dstSocket, sndBuffer2, 0x004c, 0);
if(0 > ret){
    printf("Socket Send Error¥n");
    return -1;
}

////////////////////////////////////////
// Work ID Done Notify
////////////////////////////////////////

// 0x10010008 Notify Recieve <- SC-20
ret = recv(dstSocket, rcvBuffer, 4096, 0);
if(0 >= ret){
    printf("Socket Recieve Error¥n");
    return -1;
}
int len = 0;
printf("¥n-------------------------------------------------------¥n");
printf("@ Work ID Done Notify¥n");
```

```
    printf("-------------------------------------------------------------¥n¥n");
    printf("MsgID      : 0x%02X%02X%02X%02X¥n", rcvBuffer[3], rcvBuffer[2], rcvBuffer[1], rcvBuffer[0]);
    printf("TermID     : %d¥n", ((unsigned int)rcvBuffer[7] << 24) | ((unsigned int)rcvBuffer[6] << 16) | ((unsigned
int)rcvBuffer[5] << 8) | ((unsigned int)rcvBuffer[4]));
    strcpy(tmpBuffer, &rcvBuffer[8]);
    printf("TermName   : %s¥n", tmpBuffer);
    printf("Date       : %04d/%02d/%02d %02d:%02d:%02d¥n", char2short(&rcvBuffer[0x48])
                                                                    , rcvBuffer[0x4A]
                                                                    , rcvBuffer[0x4B]
                                                                    , rcvBuffer[0x4C]
                                                                    , rcvBuffer[0x4D]
                                                                    , rcvBuffer[0x4E]
    );
    strcpy(tmpBuffer, &rcvBuffer[0x0050]);
    printf("WorkID     : %s¥n", tmpBuffer);

    // 0x00010008 Work ID Done Notify Response Send -> SC-20
    memset(sndBuffer2, 0, 0x004C);
    sndBuffer2[0x0000] = 0x08;
    sndBuffer2[0x0001] = 0x00;
    sndBuffer2[0x0002] = 0x01;
    sndBuffer2[0x0003] = 0x00;
    sndBuffer2[0x0004] = (char)((termID & 0x000000FF) >>  0);
    sndBuffer2[0x0005] = (char)((termID & 0x0000FF00) >>  8);
    sndBuffer2[0x0006] = (char)((termID & 0x00FF0000) >> 16);
    sndBuffer2[0x0007] = (char)((termID & 0xFF000000) >> 24);
    sprintf(&sndBuffer2[0x0008], termName);
    ret = send(dstSocket, sndBuffer2, 0x0048, 0);
    if(0 > ret){
        printf("Socket Send Error¥n");
        return -1;
    }

    // Close Scocket
    closesocket(dstSocket);
    closesocket(srcSocket);

    // Windows Only
    WSACleanup();

    return 0;
}
```

# 6. Flowchart


Start
Prepare the other device (such as the local PC), SC-20 SOCKET, sockaddr_in
Execute WSAStartup() (Unique to Windows)
Execute socket() Other device-side socket generation
Execute bind() Other device-side socket bind
Execute listen() Other device-side socket listen
Execute accept() Reception standby of SC-20-side socket in the other device-side socket that is set
SC-20-side socket connection completion and generation (The SC-20 Socket Mode must be set to enabled)
Job ID execution request 0x00000005 data creation
Execute send() 0x00000005 data transmission
Execute recv() Job ID execution response 0x10000005 reception
Execute recv() Inspection step completion matching notification 0x10000002 reception
Inspection step completion response 0x00010007 data creation
Execute send() 0x00010007 data transmission
Execute recv() Job ID completion notification 0x10010008 reception
Job ID completion response 0x00010008 data creation
Execute send() 0x00010008 data transmission
Execute closesocket() Socket release of both the socket at the other device-side and the SC-20-side socket
Execute WSACleanup() (Unique to Windows)
End

## Version History

| Ver. | Date | Changes | Note |
|---|---|---|---|
| 1.0 | 14th June, 2023 | New issue | |
| | | | |
| | | | |