Usage of Type0025 Module

Rev. 1.4

June 9, 2023

1 Introduction

This document describes supplement things to use the module. Some of these are restriction of the current version module.

2 Supported camera

The Type0025 module supports Z 50. The module cannot control two or more cameras, can control one camera only.

3 Environment

For the operating environment of Module SDK, refer to the following chapter of ReadMe_Eng.txt.

- Environment of operation

4 About applications using Module SDK

We recommend that the application using Module SDK to use the development environment in which Module SDK is created.

The development environment of the sample program and Module SDK is the same. For the development environment of the sample program, refer to the following chapter of ReadMe_Eng.txt.

- Contents - [Windows] - Sample Program
- Contents - [Macintosh] - Sample Program

5 Capabilities

Client should acquire the value of each Capability once now after opening of Source object. (There is no necessity for acquiring the value every time before setting the value.) When the setting of the value is executed by kNKMAIDCommand_CapSet without acquiring the value, the value to which Client did set might not be correctly set to the camera.

5.1 kNkMAIDCapability_ProgressProc

The module notifies progress information through MAIDProgress function. When the module can't compute how much the task is finished, the module will call MAIDProgress function with ulTotal = 0 and ulDone = Non-0. When the task has finished, the module will call function with ulDone = ulTotal.

5.2 kNkMAIDCapability_EventProc

MAID3.1 specification says that the client doesn't have to set MAIDEvent function to kNkMAIDCapability_EventProc. But the current module assumes that the client always sets the MAIDEvent function. So if the client doesn't set MAIDEvent function to EventProc, there are following restrictions to use the module.

1) The client can't use kNkMAIDCommand_EnumChildren.

2) The client can't support lens exchange and device turn off and on.

3) The module doesn't notify changing of capability value, so the client should keep checking these values.

5.3 kNkMAIDCapability_Children

The client may use this capability to enumerate the child objects. The client also can use kNkMAIDCommand_EnumChildren for same purpose. If the client doesn't set MAIDEvent function to kNkMAIDCapability_EventProc, the client should use kNkMAIDCapability_Chilren to enumerate the child objects.

5.4 kNkMAIDCapability_PictureControlDataEx2, kNkMAIDCapability_MoviePictureControlDataEx2

The client will use it when setting picture control of still image or movie. For details about the data format to be used, the range of values that can be used for each data item, refer to 3.39. PictureControlDataEx2, 3.136. MoviePictureControlDataEx2, MAID3Type0025.pdf.

5.5 kNkMAIDCapability_DeleteDramImage

The timing of deletion for DRAM image is limited to the following case. The module does not support the deletion on the timing excluding the following case.

- After issuing kNkMAIDCapability_Acquire, and before issuing kNkMAIDCommand_Close for Image Object.

The example of the command sequence is shown to the following table.

No	Command/Capability/Event	Object Type
1	kNkMAIDCapability_Capture	Source

2	kNkMAIDCapability_Children	Source
3	kNkMAIDCommand_Open	Item
4	kNkMAIDCapability_Children	Item
5	kNkMAIDCommand_Open	Image
6	kNkMAIDCapability_DataProc (Set)	Image
7	kNkMAIDCapability_Acquire	Image
8	kNkMAIDCommand_Async	Image
9	kNkMAIDCommand_Abort	Image
10	kNkMAIDCapability_CurrentItemID	Source
11	kNkMAIDCapability_DeleteDramImage	Source
12	kNkMAIDCapability_DataProc (Reset)	Image
13	kNkMAIDCommand_Close	Image
14	kNkMAIDCommand_Close	Item

The execution of kNkMAIDCapability_Acquire is needed before the execution of kNkMAIDCapability_DeleteDramImage. So, in the case of small data size image, JPEG Basic, the all of image data may complete reading by the kNkMAIDCapability_Acquire before issuing of deletion command. In that case, the error doesn't occur when the deletion command is executed, but the image will be saved in client program.

When the callback function was set to kNkMAIDCapability_ProgressProc, the termination of operation will be notified with the parameter of callback function, "ulDone == ulTotal" or "ulDone == ulTotal==0". But when the client aborts the operation by kNkMAIDCommand_Abort, the termination of operation will not be notified.

5.6 kNkMAIDCapability_Capture

If you start shooting using this capability, Module does not return control until the shooting is completed on the camera side.

(except for bulb shooting)

If you want to stop the continuous shooting in the middle, or to get the camera state of being shot execution, use the kNkMAIDCapability_CaptureAsync.

5.7 kNkMAIDCapability_AFCapture

If you start shooting using this capability, Module does not return control until the AF shooting is completed on the camera side.

If you want to stop the continuous shooting in the middle, or to get the camera state of being shot execution, use the kNkMAIDCapability_AFCaptureAsync.

5.8 kNkMAIDCapability_CaptureAsync

If you start shooting by using this capability, Module returns control at the time of starting the shooting operation by the camera without waiting for shooting completion. If you want to stop the continuous shooting in the middle, or to get the camera state of being shot execution, use this capability.

For getting the result of shooting, use kNkMAIDCapability_DeviceReady.

5.9 kNkMAIDCapability_AFCaptureAsync

If you start shooting by using this capability, Module returns control at the time of starting the AF shooting operation by the camera without waiting for AF shooting completion. If you want to stop the continuous shooting in the middle, or to get the camera state of being shot execution, use this capability.

For getting the result of shooting, use kNkMAIDCapability_DeviceReady.

5.10 kNkMAIDCapability_DeviceReady

This capability is used to check the operating status of the shooting started by kNkMAIDCapability_CaptureAsync or kNkMAIDCapability_AFCaptureAsync.

5.11 kNkMAIDCapability_TerminateCapture

This capability is used for stopping the bulb shooting, for stopping the continuous shooting started by kNkMAIDCapability_CaptureAsync or kNkMAIDCapability_AFCaptureAsync.

5.12 kNkMAIDCapability_GetVideoImageEx

If you start the data acquisition partially of the movie file in kNkMAIDCapability_GetVideoImageEx, and when you perform the following operation in a state in which the acquisition of all the data in the movie file is not completed,

the movie files split acquisition information in the camera is reset and it is necessary to re-acquire from the beginning.

・Card insertion and removal

・Starting movie recording.

・Starting movie playback.

・The execution of kNkMAIDCapOperation_Set for any Capability. ・Resetting a menu or white balance data.

・The shooting with the settings including the card in kNkMAIDCapability_SaveMedia.

・Value change of kNkMAIDCapability_LockCamera.

・Value change of kNkMAIDCapability_MovRecInCardStatus

・Issuance interval of kNkMAIDCapability_GetVideoImageEx more than 60 seconds.

5.13 kNkMAIDCapability_SaveMedia

Select SDRAM when transferring images without inserting a card into the camera.

6 About SDRAM save shooting

When you run the shooting with recording media SDRAM, you must issue kNkMAIDCommand_Open about Image of Item Object to be generated under the Source Object. And you must issue kNkMAIDCapability_Acquire to get all, or issue kNkMAIDCapability_DeleteDramImage to remove.

You must close the Item Object rapidly after completion of acquired or removed, since the module cannot detect the state change of the camera during the period open for Item Object.

If you do not run the deletion or acquisition of Image, there are cases where the next shooting or later cannot be carried out successfully.

The capability available SDRAM save shooting, there is a following.

- kNkMAIDCapability_Capture
- kNkMAIDCapability_AFCapture
- kNkMAIDCapability_CaptureAsync
- kNkMAIDCapability_AFCaptureAsync
- kNkMAIDCapability_CaptureDustImage

7 About Movie file recording

After running the movie recording, the movie files are generated for each 4GB.

File format depend on kNkMAIDCapability_MovieFileType.

In addition, movie file, regardless of the setting of kNkMAIDCapability_SaveMedia, is generated on the card.

After running the movie recording, for Video of Item Object to be generated under the Source Object, you must issue kNkMAIDCommand_Open always. If you need to get the Video, you issue the kNkMAIDCapability_GetVideoImageEx. (Video acquisition not required)

You must close the Item Object rapidly after open or video acquisition, since the module cannot detect the state change of the camera during the period open for Item Object.

8 Image and Thumbnail Data

An image data file is transferred from the module through MAID Data Delivery Function. (refer to 5.27 File Data Delivery Structure and 10.3 MAID Data Delivery Function in MAID3.pdf).

All thumbnail images are raw byte data in order of RGBRGBRGB…. The pixel order is from left to right and from top to bottom. The size of thumbnail image is fixed as follows.
Width: 160 pixels Height: 120 pixels
The thumbnail image may not be acquired by the timing. (refer to 4.19. Acquire, MAID3Type0025.pdf)

9 Connection with camera

If the client sends kNkMAIDCommand_Async to the module, it can know the camera is connected with PC through AddChild event for module object. When the module detects the camera is turned off, the module sends RemoveChild event for the current opened module object.

10 Opening object

The client can open only one object at same object type(eNkMAIDObjectType). (e.g. If there are two source object with different ID, client can open either one at the same time.)

But exceptional case, image and thumbnail object, these are belong to kNkMAIDObjectType_DataObj, can be opened at the same time, from same ID Item object.

11 The restriction of bulb photography

When the client shoots bulb photography with module, the maximum exposure time is 59 minutes 59 seconds. If the client shoots bulb photography with the exposure time more than maximum exposure, shooting will not be guaranteed.

The example of the command sequence is shown to the following table.

No	Capability,Command	Precautions
1	kNkMAIDCapability_Capture	In case of bulb photography, the return value will be kNkMAIDResult_BulbReleaseBusy
2	kNkMAIDCommand_Async	Until issue TerminateCapture, the client can issue Async optionally repeatedly. The maximum time from Capture and TerminateCapture (= the maximum exposure time) is 59 minutes 59 seconds.
3	kNkMAIDCapability_TerminateCapture	The client must issue TerminateCapture within 59 minutes 59 seconds from Capture issued. If long exposure noise reduction setting is ON, see 13.2.

12 The restriction about Z 50.

12.1 Live view

The camera starts the camera live view after the power is turned on, except when the live view prohibition state is set. (The live view prohibition state can be checked with kNkMAIDCapability_LiveViewProhibit.)

In order to acquire live view data with kNkMAIDCapability_GetLiveViewImage, kNkMAIDCapability_LiveViewStatus, it is necessary to transition to the remote live view state.

The capabilities that can be set during live view refer to 10. Capability table that can be set during live view photography, movie live view, movie recording, SpotWB, in MAID3Type0025(E).pdf.

12.2 When long exposure noise reduction is ON

In case of “Long Exposure NR” is ON, the time until the image is created from the start of exposure is twice the exposure time. The module doesn’t return control until the creation of image has complete.

In case of bulb photography, image generation is started after running kNkMAIDCapability_TerminateCapture. The time to complete the generation of images from the execution of kNkMAIDCapability_TerminateCapture will need the time same as exposure time, meanwhile, the module doesn’t return control.

12.3 Auto Bracketing

Shutter speed and Aperture cannot be changed when auto bracketing is in effect.

When kNkMAIDCapability_LiveViewSelector is "1: movie live view", bracketing shooting on the camera side indicates ON, but actually the camera cannot perform bracketing shooting.

For this reason, the module sets the value of kNkMAIDCapability_EnableBracketing to "Fasle: OFF" and sets it to Read Only.

In the capability including kNkMAIDCapability_EnableBracketing as a condition such as setting prohibition, it judges by not referring to the current value of kNkMAIDCapability_EnableBracketing but the ON / OFF state of the bracketing shooting on the camera body side.

For details, refer to each chapter of MAID3Type0025(E).pdf.

- 3.53. kNkMAIDCapability_HDRMode
- 3.54. kNkMAIDCapability_HDRExposure
- 3.55. kNkMAIDCapability_HDRSmoothing

- 3.56. kNkMAIDCapability_HDRSaveIndividualImages
- 3.70. kNkMAIDCapability_ShutterSpeed
- 3.72. kNkMAIDCapability_Aperture
- 3.97. kNkMAIDCapability_BracketingCount

12.4 Depth-of-field preview button

Module may not work properly while the camera's Depth-of-field preview button is pressed.

12.5 Synchronization record of the movie

This camera cannot record movie files to an external recording device connected by HDMI.

Therefore, if you set the value of kNkMAIDCapability_MovRecInCardStatus to 1(ON) and the movie file recording starts successfully, kNkMAIDResult_RecInCard will invariably be returned.

13 The restriction on Macintosh

After connecting your camera to your PC, please wait to start module until the memory card access lamp stops flashing.

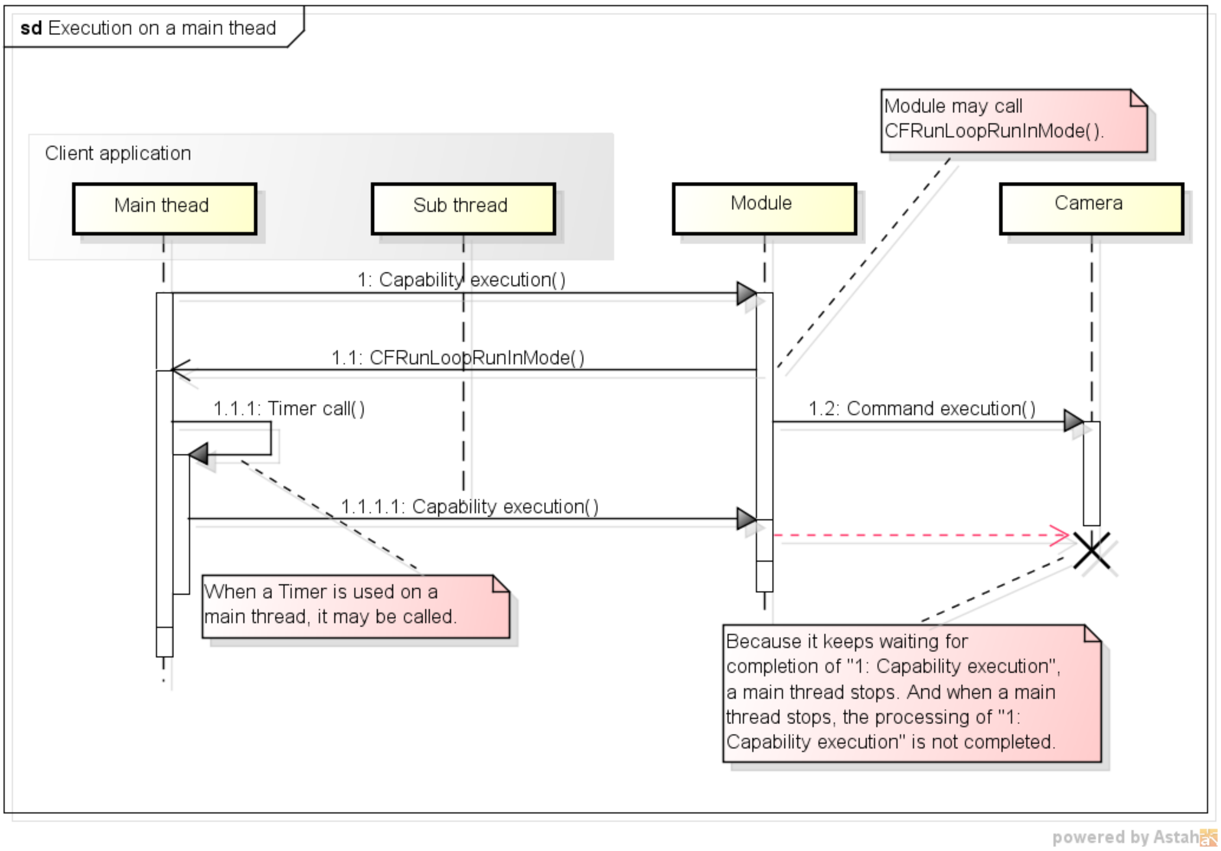
When you connect a camera set in the “Camera control” mode to the computer, the connected camera will show under the SHARED list in the Image Capture application (bundled with OS). To use Module through the LAN connection, please make sure that you do not select the camera on the list.

Client application must not stop a main thread during execution of Capability when you use Module for Macintosh.

When client stops a main thread during execution of Capability, Module may not return from processing of that Capability, because Module can't receive the response from a camera.

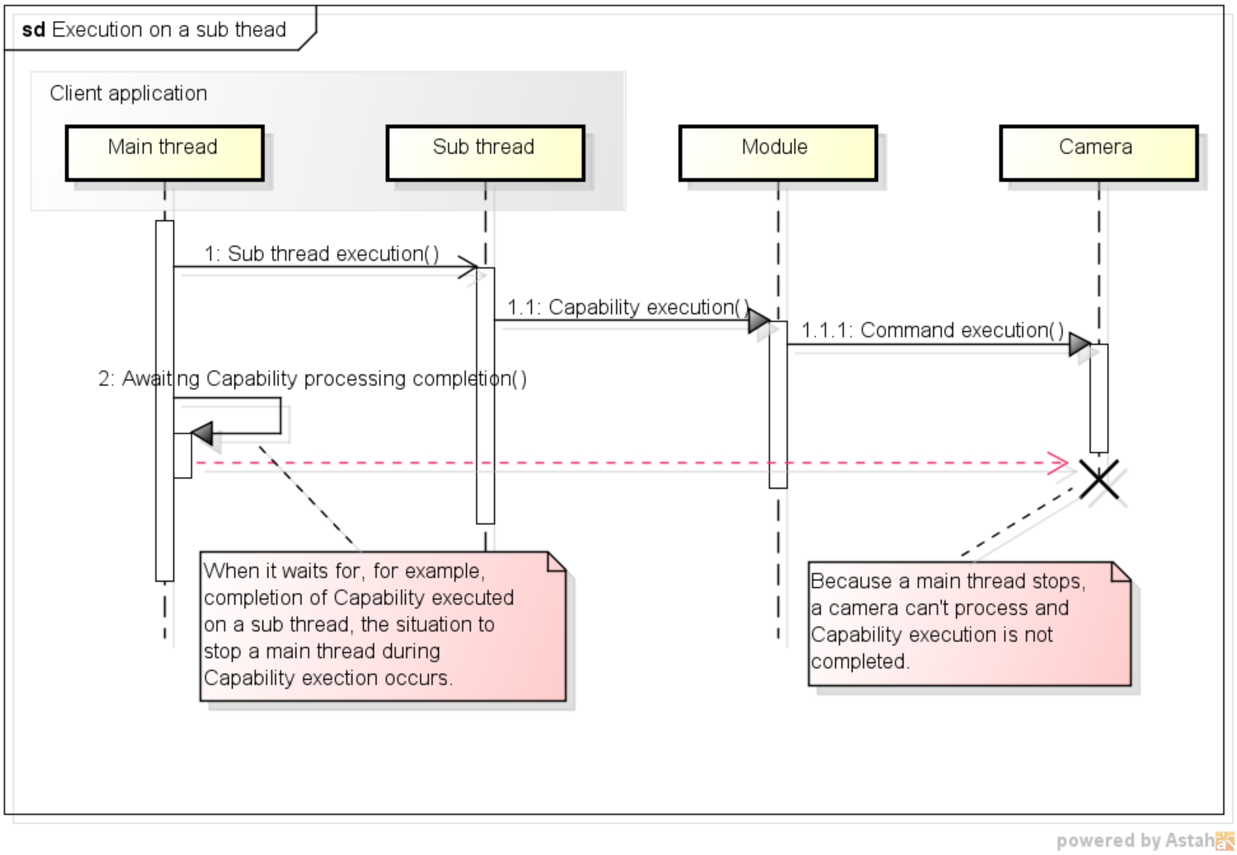
Example 1) When client application executes Capability from a main thread, Module may call CFRunLoopRunInMode(). Therefore the timer in the main thread may be called, for example, though control shifts to Module.

When the client executes other Capability at the timer processing, it may become the wait state. This is because processing of Capability which it executed from a main thread earlier is not completed. Because the timer processing is executed on a main thread, as a result, the main thread stops. Therefore the Capability that executed earlier can't receive the response from a camera and it reaches to a deadlock state.


sd Execution on a main thead
Client application
Main thead
Sub thread
Module
Camera
Module may call CFRunLoopRunInMode().
1: Capability execution()
1.1: CFRunLoopRunInMode()
1.1.1: Timer call()
1.2: Command execution()
1.1.1.1: Capability execution()
When a Timer is used on a main thread, it may be called.
Because it keeps waiting for completion of "1: Capability execution", a main thread stops. And when a main thread stops, the processing of "1: Capability execution" is not completed.
powered by Astah

Example 2) When client application may execute Capability from a sub thread and wait for the processing completion of Capability in main thread, client must not stop the main thread.

In this case, client has to call CFRunLoopRunInMode()at fixed intervals until the completion of Capability.


sd Execution on a sub thead
Client application
Main thread
Sub thread
Module
Camera
1: Sub thread execution()
1.1: Capability execution()
1.1.1: Command execution()
2: Awaiting Capability processing completion()
When it waits for, for example, completion of Capability executed on a sub thread, the situation to stop a main thread during Capability exection occurs.
Because a main thread stops, a camera can't process and Capability execution is not completed.
powered by Astah

14 Structure Member Alignment

In MAID3.H, there is a comment saying that all alignments are 4byte, but this value depends on platform.

15 History

- Rev.1.4　June 9, 2023
  - 13.1 Important note about macOS 10.15…Deleted.

- Rev.1.3　October 28, 2022
  - 13. The restriction on Macintosh…Deleted.

- Rev.1.2　October 29, 2021
  - 3. Environment…Update the environment of Windows and Macintosh.
  - 4. About applications using Module SDK...Update the development environment.
  - 5. Runtime Library…Delete.
  - 12.3 Auto Bracketing…Update.

- Rev.1.1　February 26, 2020
  - 3. Environment...Update the environment of Windows and Macintosh.
  - 14.1. Important note about macOS 10.15...Added.

- Rev.1.0　November. 1, 2019　First version