|  |  |
| --- | --- |
| **(Confidential)** | |
| Scope of disclosure | [Company] for EPH only |
| Period of confidentiality | 7 year after issued date |
| Head of Information Owner | Head of engineer department |
| Handling restriction | NA |

**Software Detailed Design Specification**

**EPH**

**Toshiba Software Development (Vietnam) Co., Ltd.**

|  |
| --- |
| Document ID: TSDV-21A-EPH-SDD |
| Total: 16 Page No. 1 |

Revision History

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Rev. No.  (X.YY) | Date (YYYY-MM-DD) | Section No. Changed | Change Description | Author | Reviewed by | Approved by |
| 0.01 | 2021-09-09 | All | Initialize document | HoaDV |  |  |
| 0.02 | 2021-09-10 | 5 | Add sequence diagrams | HoaDV |  |  |
| 0.03 | 2021-09-13 | 4,5 | Update classes diagram to add parameters for operation.  Update sequence of GetImageRecognizeResult to remove unnecessary step. | HoaDV |  |  |
| 0.04 | 2021-09-14 | 5 | Add table for step description of sequences.  Fix missing step (of register pariticipant) for sequence of GetImageRecognizeResult and RecognizeImage | HoaDV | ThanhBX |  |
| 0.05 | 2021-09-17 | 4,5 | Update for team review comment | HoaDV |  |  |
| 0.06 | 2021-10-05 | 4 | Add class descriptions | HoaDV |  |  |
| 0.07 | 2021-10-12 | 6 | Add chapter 6 for describe API spec | ThaiNH | HoaDV |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Content

[1 Introduction 6](#_Toc85010337)

[*2* References 6](#_Toc85010338)

[3 Definitions and Acronyms 6](#_Toc85010339)

[4 Classes Detailed Design 7](#_Toc85010340)

[4.1 Interface Design 7](#_Toc85010341)

[4.2 Internal Classes Detail Design 8](#_Toc85010342)

[5 Sequences Detailed Design 10](#_Toc85010343)

[5.1 WorkerThread Detailed Design 10](#_Toc85010344)

[5.2 SaveCapturedImage Job Detailed Design 12](#_Toc85010345)

[5.3 CaptureImage Job Detailed Design 13](#_Toc85010346)

[5.4 GetImageRecognizeResult Job Detailed Design 14](#_Toc85010347)

[5.5 RecognizeImage Job Detailed Design 15](#_Toc85010348)

[5.6 Stop SDK Detailed Design 16](#_Toc85010349)

[6 API specification 17](#_Toc85010350)

[6.1 WorkerThread 17](#_Toc85010351)

[6.1.1 start 17](#_Toc85010352)

[6.1.2 stop 17](#_Toc85010353)

[6.1.3 join 18](#_Toc85010354)

[6.1.4 stopAndWait 18](#_Toc85010355)

[6.1.5 submitJob 18](#_Toc85010356)

[6.1.6 submitPeriodicJob 19](#_Toc85010357)

[6.1.7 removePeriodicJob 19](#_Toc85010358)

[6.1.8 removePeriodicJob 20](#_Toc85010359)

[6.1.9 getMessage 20](#_Toc85010360)

[6.1.10 process 21](#_Toc85010361)

[6.1.11 isBusy 21](#_Toc85010362)

[6.1.12 getLastProcessingTime 22](#_Toc85010363)

[6.2 SaveCaptureImage Job 23](#_Toc85010364)

[6.2.1 buildFilePathName 23](#_Toc85010365)

[6.2.2 saveImage 24](#_Toc85010366)

[6.2.3 readAndSaveImage 24](#_Toc85010367)

[6.3 CaptureImage Job 25](#_Toc85010368)

[6.3.1 catureImage 25](#_Toc85010369)

[6.3.2 convertToJPEG 25](#_Toc85010370)

[6.3.3 publishImage 26](#_Toc85010371)

[6.3.4 allocPublishData 26](#_Toc85010372)

[6.4 RecognizeImage Job 27](#_Toc85010373)

[6.4.1 recognizeImage 27](#_Toc85010374)

[6.4.2 sendRecognitionResponse 27](#_Toc85010375)

[6.4.3 createResultData 28](#_Toc85010376)

[6.5 GetImageRecognizeResult Job 28](#_Toc85010377)

[6.5.1 readImage 28](#_Toc85010378)

[6.5.2 receiveResult 29](#_Toc85010379)

[6.5.3 publishRecognizeImage 29](#_Toc85010380)

[6.5.4 allocPublishData 30](#_Toc85010381)

[6.6 DnnRecognition 31](#_Toc85010382)

[6.6.1 queueDnnJob 31](#_Toc85010383)

[6.6.2 start 31](#_Toc85010384)

[6.6.3 stop 32](#_Toc85010385)

[6.6.4 join 32](#_Toc85010386)

[6.6.5 stopAndWait 33](#_Toc85010387)

[6.6.6 readReponse 33](#_Toc85010388)

[6.6.7 writeRequest 34](#_Toc85010389)

[6.6.8 dequeueBuffer 34](#_Toc85010390)

[6.6.9 callCallbackWithResult 35](#_Toc85010391)

[6.6.10 process 35](#_Toc85010392)

List of Figures

Figure 4‑1: SDK interface design 7

Figure 4‑2: SDK internal design 8

Figure 5‑1: Sequence of job requests handling of WorkerThread 10

Figure 5‑2: Sequence of image receiving over DDS network 12

Figure 5‑3: Sequence of image capturing and sending over DDS network 13

Figure 5‑4: Sequence of requesting to recognize image over DDS network 14

Figure 5‑5: Sequence of responding to image recognition request over DDS network 15

Figure 5‑6: Sequence of stop SDK 16

# Introduction

This document describes detail design for EPH’s SDK of 2021A. This design only covers functionalities that describes clearly in SRS document.

Because quality level of project is prototyping in only 1 month, the design will not be described too much detail. Example, there will be no activity diagram for internal functions. The design only describes role and contribution of those child function to parent function processing.

# References

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document ID** | **Document/Standards Name/Title** | **Source** | **Version No. /Release or Publication date** | **Brief Description/Section Reference** |
| TSDV-EPH-SRS | TSDV-21A-EPH-SRS.docx | - | 0.07 | Software requirement specification for EPH SDK |
| TSDV-21A-EPH-SAD | TSDV-21A-EPH-SAD.docx | - | 0.03 | Software Architecture Design for EPH SDK |
| QnA list | TSDV-EPH-QA\_20210820.xls | - | 20210817 | Question and answer between TSDV and TTEC |
| - | - | Debug\_HPF\_evaluation\_dummy\_Reliable.zip | 20210813 | Sample source code from TTEC about using DDS with EPH board |

# Definitions and Acronyms

|  |  |  |
| --- | --- | --- |
| **No** | **Acronyms** | **Definition** |
| **1** | DDS | Data Distribution Service, a Pub/Sub technology for ubiquitous, polyglot, efficient and secure data sharing |
| 2 | EPH | EnPoint Hub, a HW device that TTEC provides to TSDV to run source code (It is used to name project also) |
| 3 | SDK | Software Development Kit |
| 4 | DNN | Deep Neural Network (for image recognition) |
| 5 | API | Application Programming Interface |

# Classes Detailed Design

## Interface Design

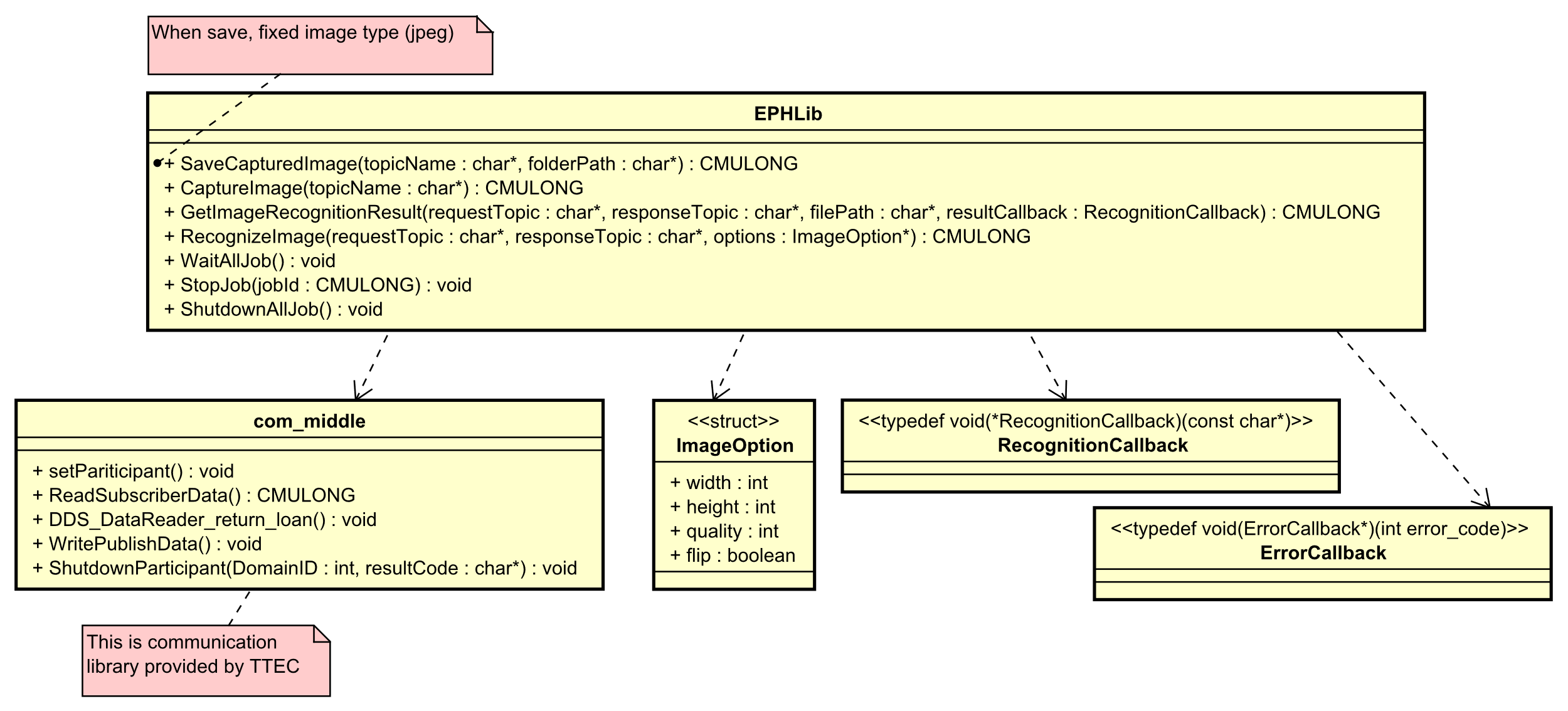


Figure 4‑1: SDK interface design

There are 4 main APIs corresponding for image transferring and recognition over DDS network of SDK.

|  |  |  |
| --- | --- | --- |
| **No.** | **API** | **Target Function** |
| 1 | SaveCapturedImage | This API is for image transferring. Application uses this API to receive image over DDS network. |
| 2 | CaptureImage | This API is for image transferring. Application uses this API to capture and send image over DDS network. |
| 3 | GetImageRecognitionResult | This API is for image recognition. Application uses this API to request to recognize image over DDS network. |
| 4 | RecognizeImage | This API is for image recognition. Application uses this API to serve image recognition requests. |

All interface functions should be exported as C-language function for application usage.

“com\_middle” is a library provided by TTEC. Then it is not target of development. But SDK uses that kind of library.

## Internal Classes Detail Design

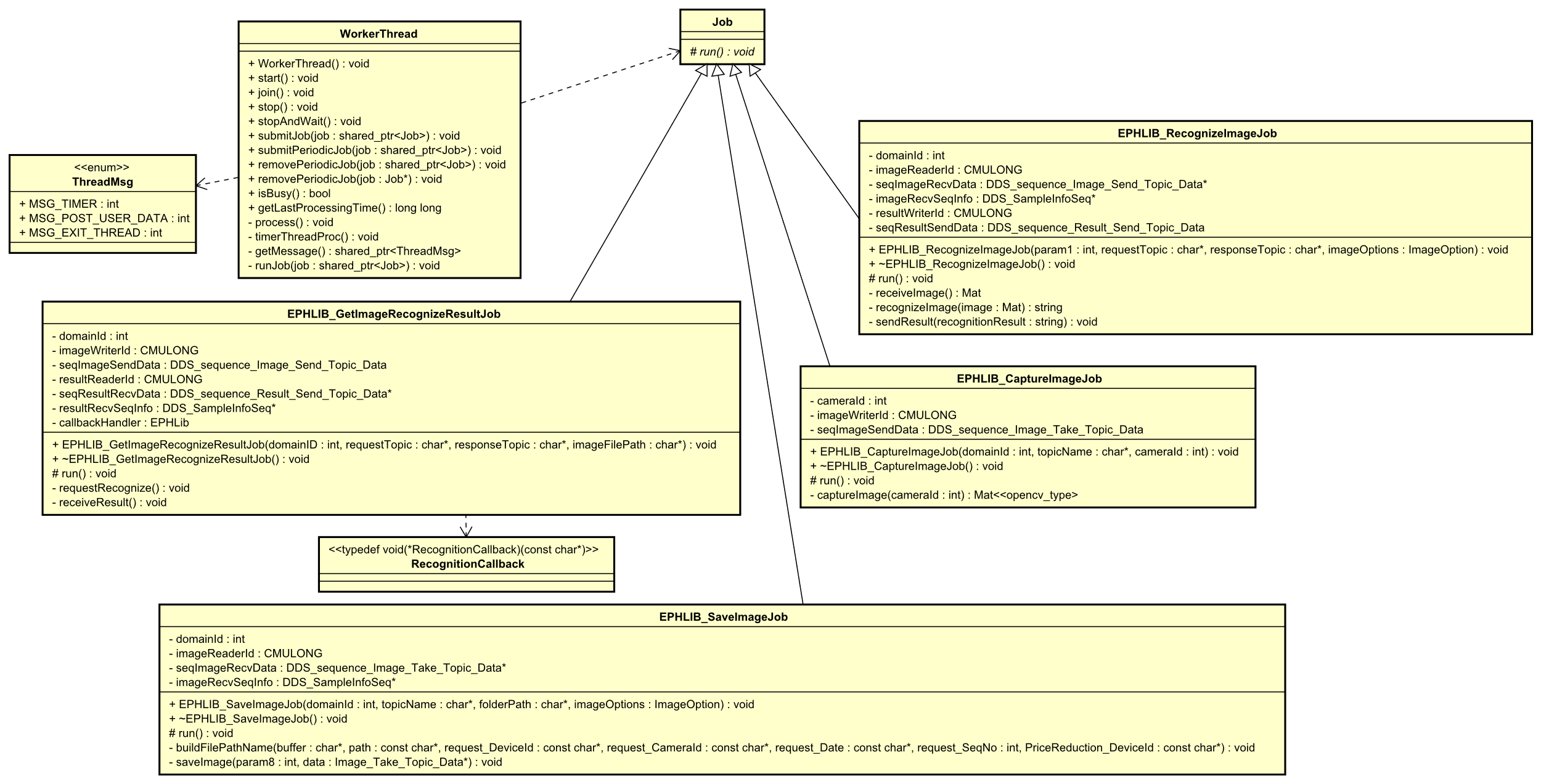


Figure 4‑2: SDK internal design

To be easy for development and scaling up later, each SDK job is designed as a separated class.

* For development, developer can develop in parallel. Since each job is handled by a class.
* Testing and debugging is also become easier because each class handle a job.
* In future, if more jobs are required to add to SDK, it is easy just to create new classes for new jobs without impact to existed ones.

The role of each class will be shown clearly in section 5 below, when interaction between classes is described.  
Role of each class is described in table below.

|  |  |
| --- | --- |
| **Class** | **Role** |
| WorkerThread | This class handle of executing job according to requests from interface API.  Base on triggered event, it will execute job accordingly. There are 3 types of event as shown in ThreadMsg data type.   * MSG\_TIMER: WorkerThread will execute all periodically jobs (in job que) when this event comes. * MSG\_POST\_USER\_DATA: WorkerThread will a job (in job que) when this event comes. After that, executed job is removed from que. * MSG\_EXIT\_THREAD: if this event comes, WorkerThread will finish its processing. |
| Job | This is abstraction class for job type. |
| EPHLIB\_SaveImageJob | This class is implementation for job that receives and saves image over DDS network. |
| EPHLIB\_CaptureImageJob | This class is implementation for job that captures and sends image over DDS network. |
| EPHLIB\_GetImageRecognizeResultJob | This class is implementation for job that requests to recognize an image and receives back result over DDS network. |
| EPHLIB\_RecognizeImageJob | This class is implementation for job that receives image, recognize image, then send back result over DDS network. |

# Sequences Detailed Design

This section describes interaction design between classes to show how requirement are realized in SDK.

There are 5 sequences with purpose of each is as below.

|  |  |
| --- | --- |
| **Section** | **Sequence Purpose** |
| 5.1 | Describes job requests handling from other modules of WorkerThread |
| 5.2 | Describes image receiving over DDS network |
| 5.3 | Describes image sending over DDSK network |
| 5.4 | Describes image recognition requesting over DDS network |
| 5.5 | Describes image recognition responding over DDS network |

## WorkerThread Detailed Design

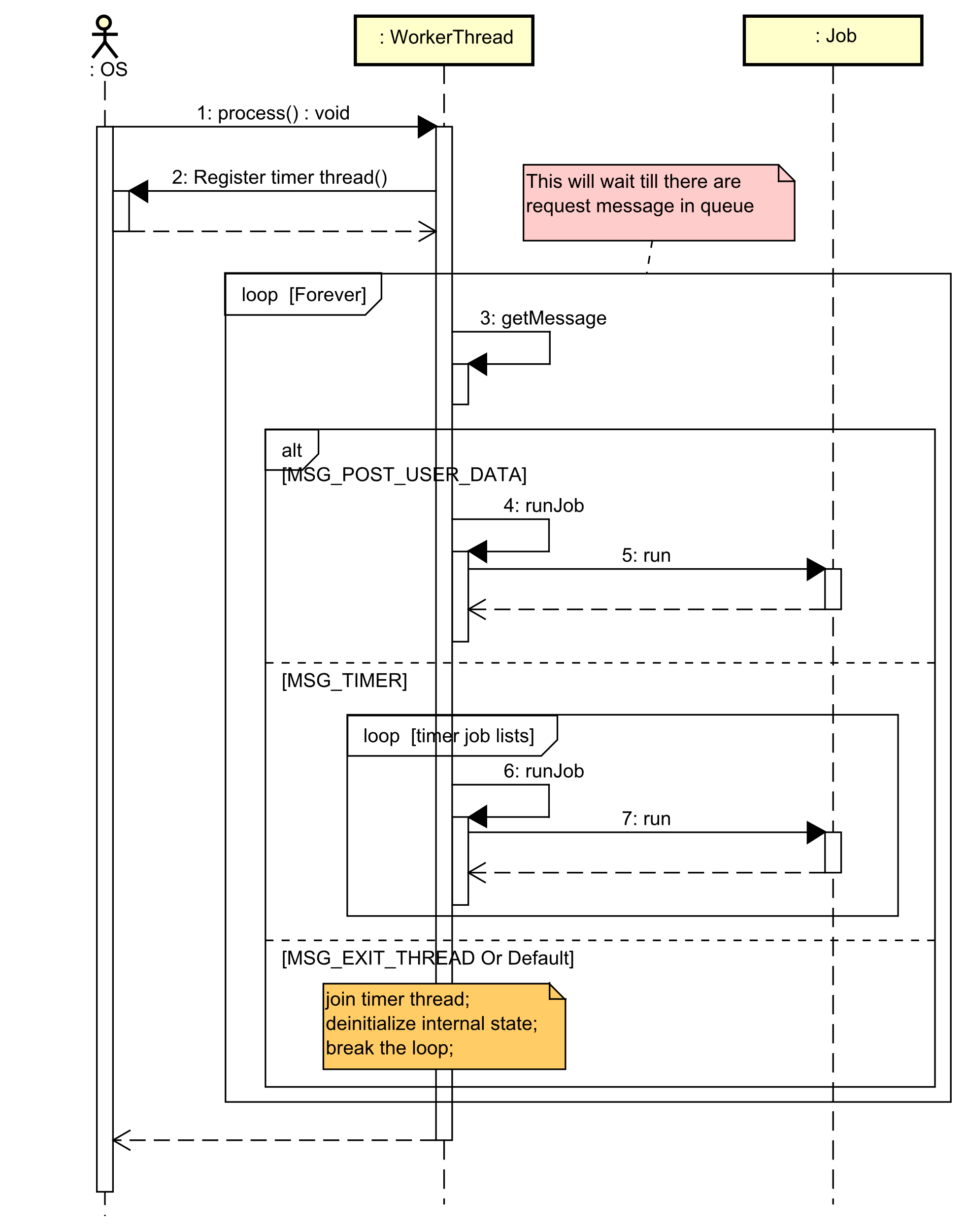


Figure 5‑1: Sequence of job requests handling of WorkerThread

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | OS calls process method of WorkThread (after the thread is created). |
| 2 | Creates a thread to acts as timer (this thread will issue a message for every 30ms). |
| - | Enters to infinitive loop to execute jobs according to coming events. |
| 3 | Gets event from event queue. If there is no event, the thread will sleep till an event comes. |
| 4, 5 | If event is a request to execute a non-periodic job, then pops 1 job from non-periodic job queue and runs job. |
| 6, 7 | If event is from timer thread, then runs all jobs from periodic job list. |
| - | If event is a request to terminate, then terminates thread by breaking infinitive loop and returns process. |

## SaveCapturedImage Job Detailed Design

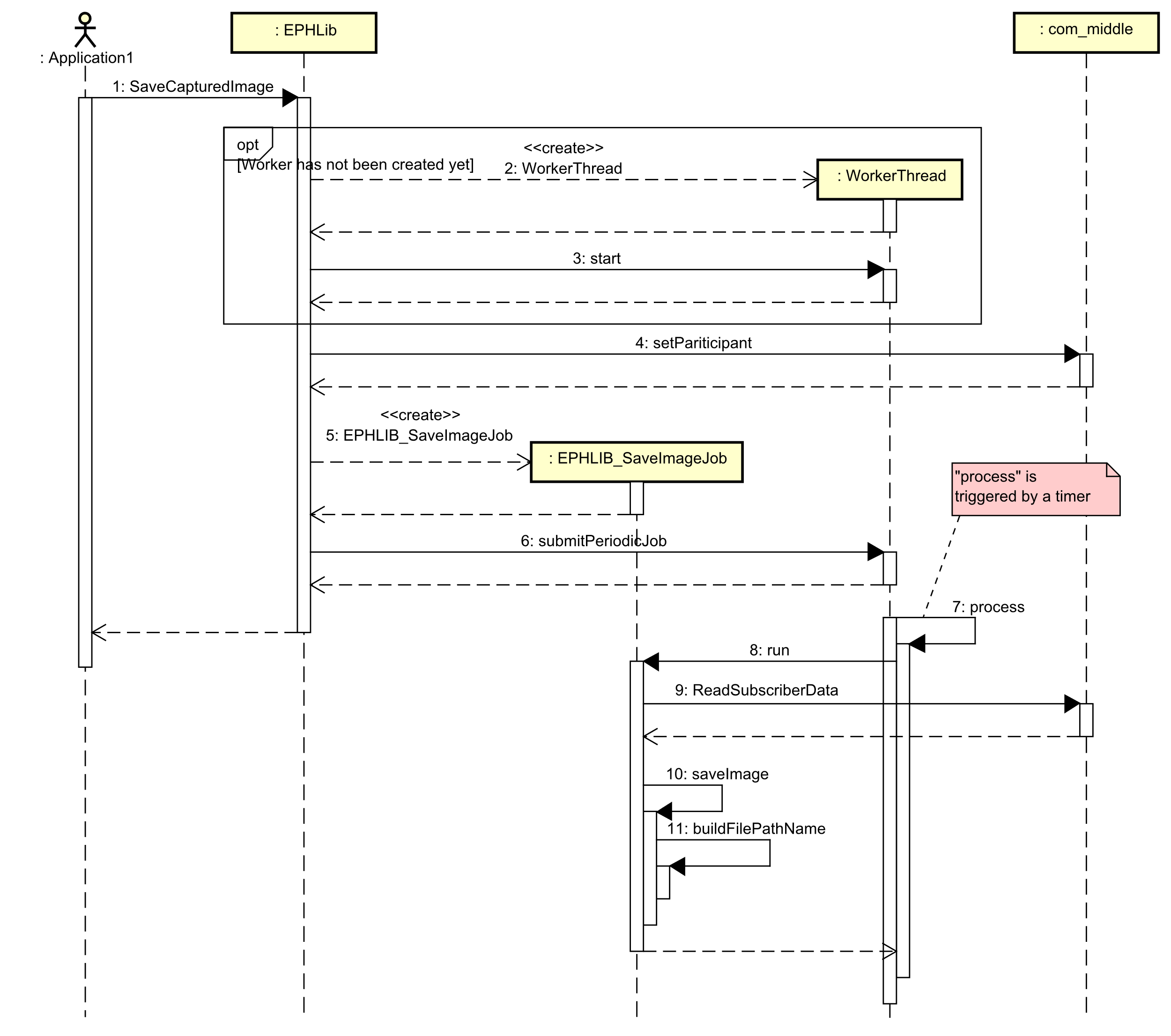


Figure 5‑2: Sequence of image receiving over DDS network

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Application calls API to start job. |
| 2, 3 | Creates and starts worker thread if have not been created before. |
| 4 | Registers communication participant ID. |
| 5, 6 | Creates and submits job (as periodic job) to worker thread. |
| 7, 8 | Worker thread runs job when cyclic event comes. |
| 9 | Reads image data from DDS network (with participant ID register in step 4 above). |
| 10, 11 | If receive image data, then saves data to image file to memory. |

## CaptureImage Job Detailed Design

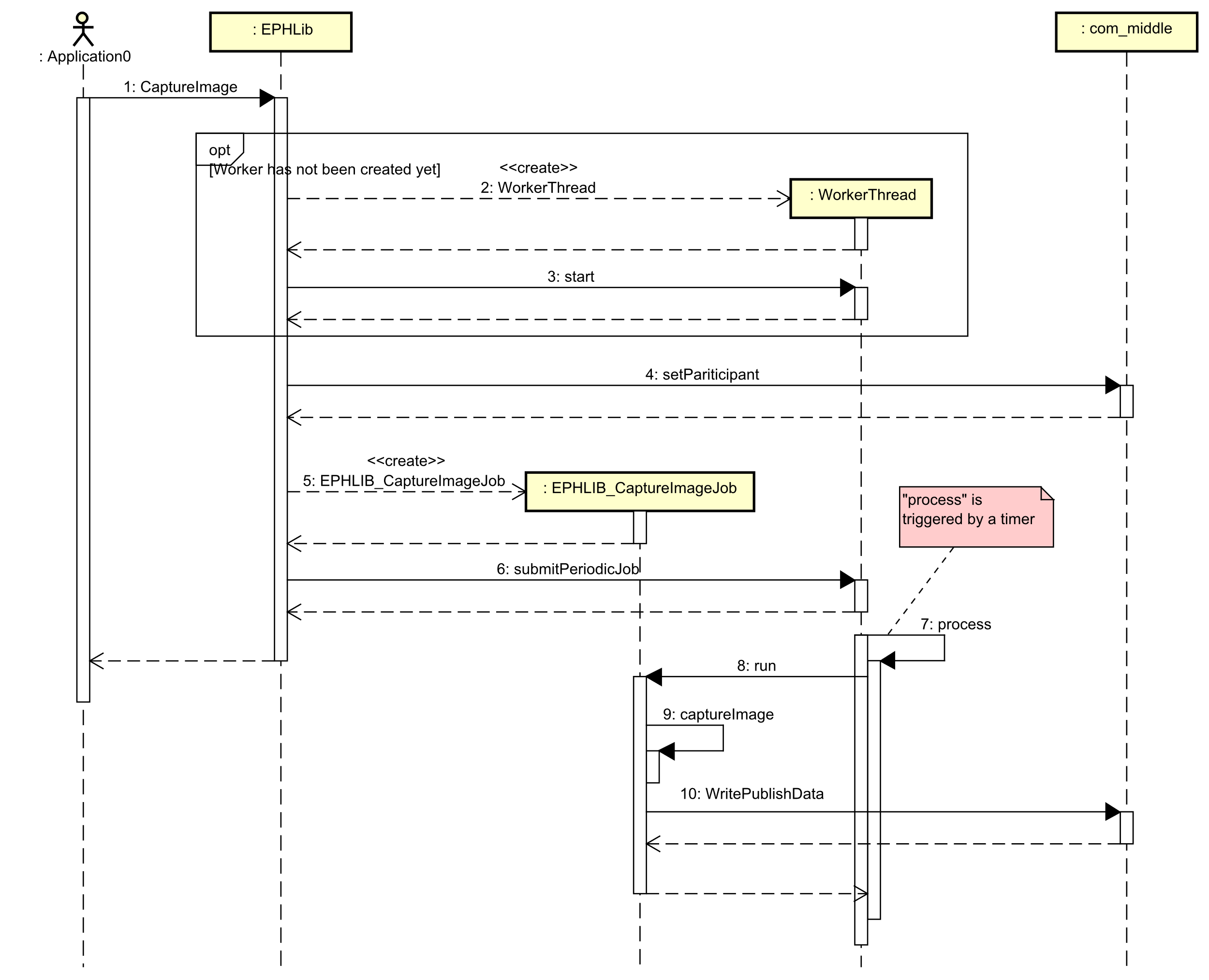


Figure 5‑3: Sequence of image capturing and sending over DDS network

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Application call API to start job. |
| 2, 3 | Creates and starts worker thread if have not been created before. |
| 4 | Registers communication participant ID. |
| 5, 6 | Creates and submit job (as periodic job) to worker thread. |
| 7, 8 | Worker thread run job when cyclic event comes. |
| 9 | Captures an image from camera. |
| 10 | Sends image data over DDS network (with participant ID registered in step 4 above). |

## GetImageRecognizeResult Job Detailed Design

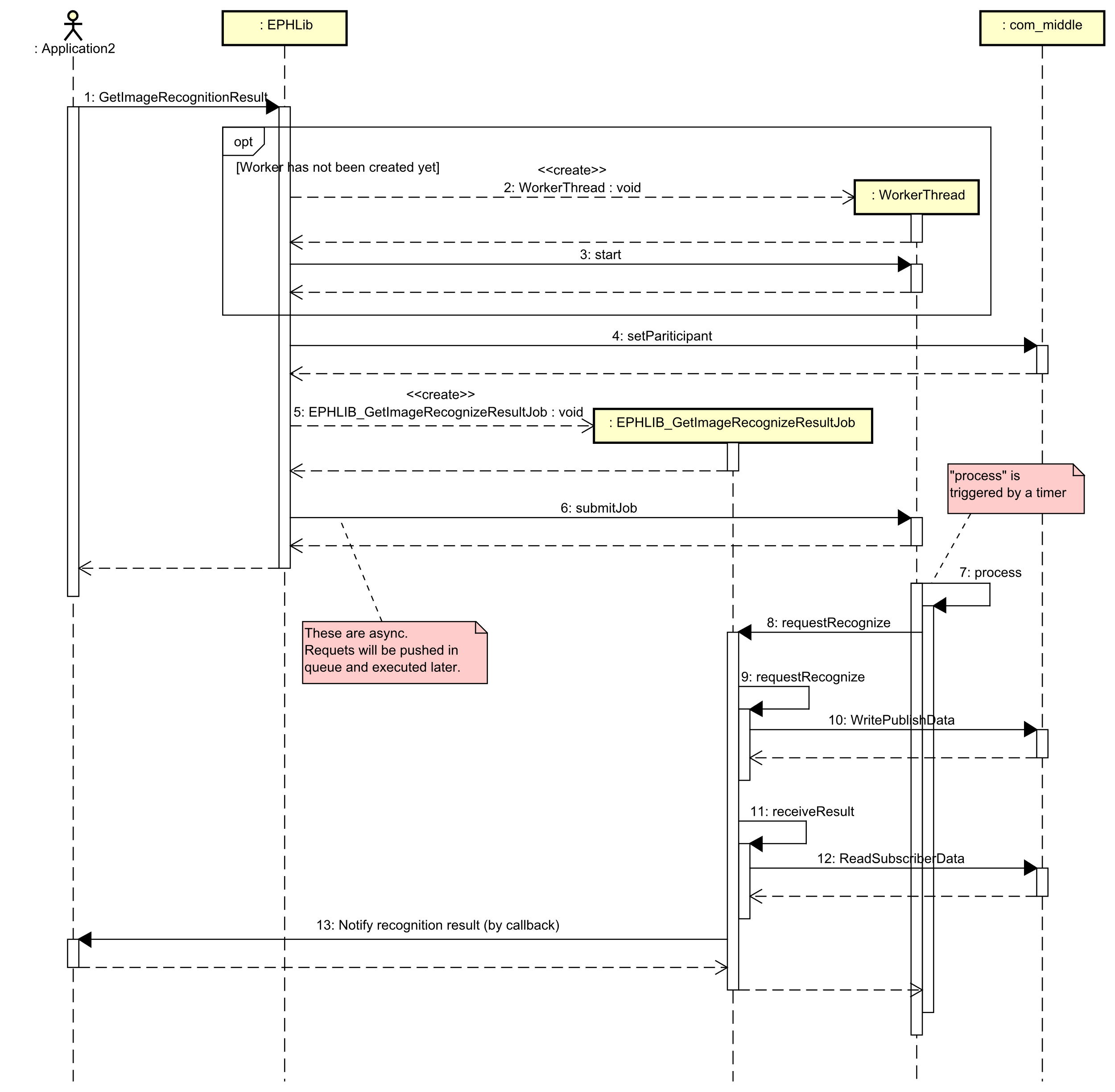


Figure 5‑4: Sequence of requesting to recognize image over DDS network

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Application calls API to start job. |
| 2, 3 | Creates and starts worker thread if have not been created before. |
| 4 | Registers communication participant ID. |
| 5, 6 | Creates and submits job to worker thread. |
| 7, 8 | Worker thread runs job when cyclic event comes. |
| 9, 10 | Sends image to be recognized over DDS network (with participant ID registered in step 4 above) |
| 11, 12 | Waits and receives recognition results over DDS network (with participant ID registered in step 4 above) |
| 13 | If receives recognition result, callbacks to application to notify result. |

## RecognizeImage Job Detailed Design

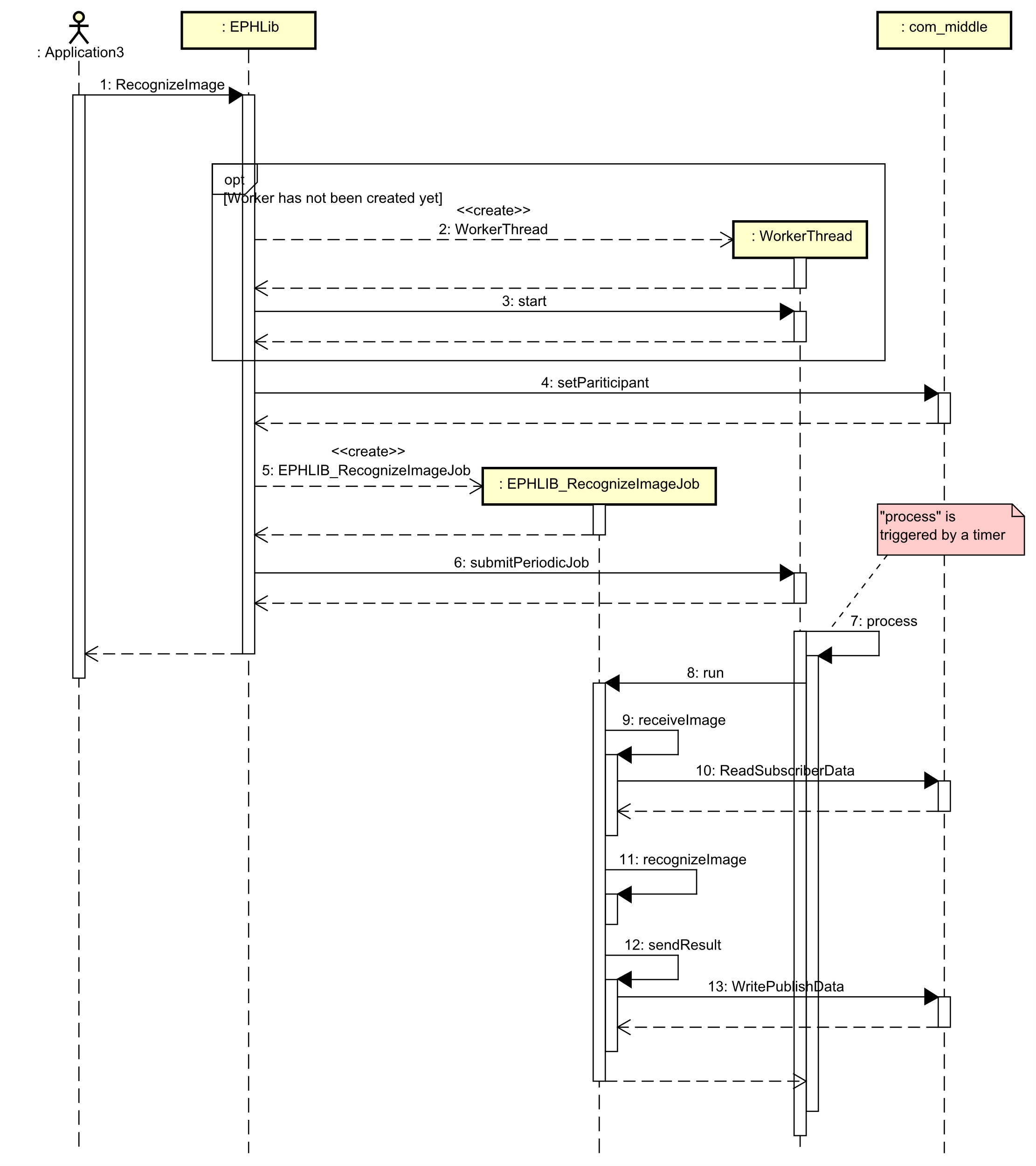


Figure 5‑5: Sequence of responding to image recognition request over DDS network

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Application calls API to start job. |
| 2, 3 | Creates and starts worker thread if have not been created before. |
| 4 | Registers communication participant ID. |
| 5, 6 | Creates and submits job (as periodic job) to worker thread. |
| 7, 8 | Worker thread runs job when cyclic event comes. |
| 9, 10 | Waits and receives image from DDS network (with participant ID registered in step 4 above). |
| 11 | Recognizes received image with DNN module. |
| 12 | Sends back recognition result over DSS network (with participant ID registered in step 4 above). |

## Stop SDK Detailed Design

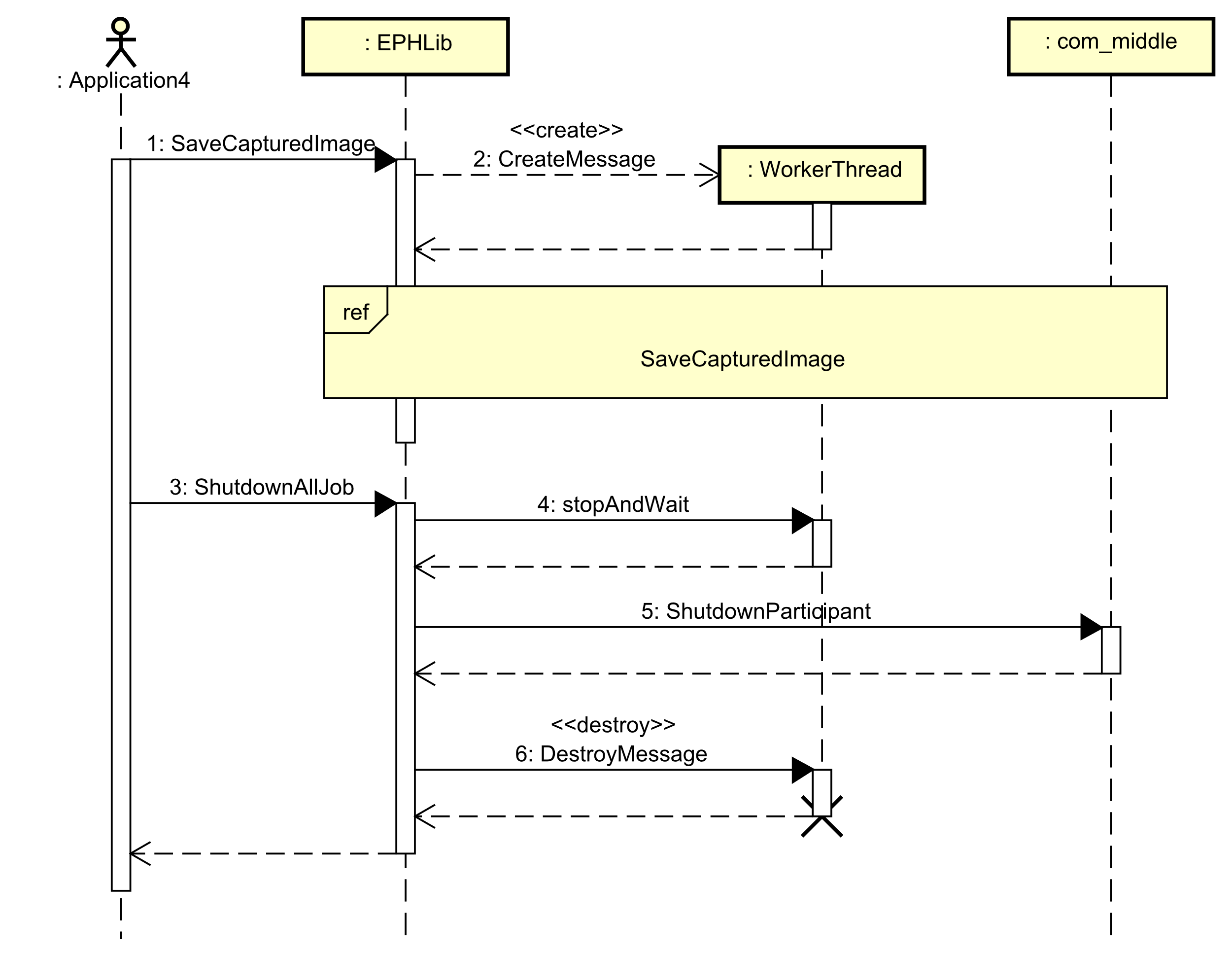


Figure 5‑6: Sequence of stop SDK

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1, 2 | Application calls API to start a job  Example: call SaveCaptureImage to receiving image over DDS network |
| 3 | Application calls ShutDownAllJob to stop SDK |
| 4 | Stops all registered jobs |
| 5 | Stop DDS communication service |
| 6 | Delete created worker thread |

# API specification

## WorkerThread

### start

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Start worker thread | | start() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when starting worker thread. | | | | | | | | | |
| Include | | | worker\_thread.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | Creates and starts a worker thread if thread has not started.  Otherwise, it returns without doing anything. | | | | | | | | | | |

### Join

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Join worker thread | | join() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when waiting until thread completes its execution. | | | | | | | | | |
| Include | | | worker\_thread.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | If worker thread has started, then blocks current caller thread and waits until worker thread finishes its execution.  Otherwise, it returns without doing anything. | | | | | | | | | | |

### stop

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Stop worker thread | | stop() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when stopping worker thread. | | | | | | | | | |
| Include | | | worker\_thread.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | If worker thread has started, then creates an exit message and notify worker thread.  Otherwise, it returns without doing anything. | | | | | | | | | | |

### stopAndWait

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Stop and wait worker thread | | stopAndWait() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when stopping all job in worker thread. | | | | | | | | | |
| Include | | | worker\_thread.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | It combines behavior of stop() and join(). | | | | | | | | | | |

### submitJob

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name(Call name) | | | | Function attributes | | Created date | Author |
| Submit Job | | submitJob() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when adding a job once time to worker thread. | | | | | | | | | |
| Include | | | worker\_thread.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | job | | job | | shared\_ptr<Job> | N/A | I | | ・job  ・valid: not NULL  ・invalid: NULL | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | Creates a request of executing job, add request to internal queue and trigger internal processing thread. | | | | | | | | | | |

### submitPeriodicJob

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Submit Periodic Job | | submitPeriodicJob() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when adding a job periodically to worker thread. | | | | | | | | | |
| Include | | | worker\_thread.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | job | | job | | shared\_ptr<Job> | N/A | I | | ・job  ・valid: not NULL  ・invalid: NULL | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | Add job to internal periodic list. | | | | | | | | | | |

### removePeriodicJob

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | Function attributes | | Created date | Author |
| Remove Periodic Job | | removePeriodicJob() | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when stopping job periodically to worker thread | | | | | | | | |
| Include | | | worker\_thread.h | | | | | | | | |
|  | | Variable name | | Name | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | job | | job | shared\_ptr<Job> | N/A | I | | ・job  ・valid: not NULL  ・invalid: NULL | | |
| Return value | | N/A | | N/A | N/A | N/A | N/A | | N/A | | |
| Processing content | | Remove a job from internal periodic job list. | | | | | | | | | |

### removePeriodicJob

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | Function attributes | | Created date | Author |
| Remove Periodic Job | | removePeriodicJob() | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when stoping job periodically to worker thread | | | | | | | | |
| Include | | | worker\_thread.h | | | | | | | | |
|  | | Variable name | | Name | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | job | | job | Job\* | 8 | I | | ・job  ・valid: not NULL  ・invalid: NULL | | |
| Return value | | N/A | | N/A | N/A | N/A | N/A | | N/A | | |
| Processing content | | Remove job from internal periodic job list. | | | | | | | | | |

### getMessage

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | | Call name | | | Function attributes | | Created date | Author |
| Get message thread | | | getMessage() | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when getting message thread in runtime and return content message found. | | | | | | | | | |
| Include | | | worker\_thread.h | | | | | | | | | |
|  | | Variable name | | Name | Data type | | sizeﾞ | I/O | | Contents | | |
| Argument | | job | | job | Job\* | | N/A | I | | ・job need removed | | |
| Return value | | N/A | | Return value | shared\_ptr<ThreadMsg> | | N/A | O | | ・Return value  ・Message thread 1: MSG\_TIMER  ・Message thread 2: MSG\_POST\_USER\_DATA  ・Message thread 3: MSG\_EXIT\_THREAD | | |
| Processing content | | Waits until message queue has at least 1 element, then get a message from queue and return. | | | | | | | | | | |

### timerThreadProc

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | | Call name | | | Function attributes | | Created date | Author |
| Set timer thread process | | | timerThreadProc() | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when setting timer interval for periodic thread processing. | | | | | | | | | |
| Include | | | worker\_thread.h | | | | | | | | | |
|  | | Variable name | | Name | Data type | | sizeﾞ | I/O | | Contents | | |
| Argument | | N/A | | N/A | N/A | | N/A | N/A | | N/A | | |
| Return value | | N/A | | N/A | N/A | | N/A | N/A | | N/A | | |
| Processing content | | Sleeps for timer interval then put a MSG\_TIMER into queue and notify worker thread. | | | | | | | | | | |

### runJob

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name(Call name) | | | | Function attributes | | Created date | Author |
| Run Job | | runJob() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when executing a job. | | | | | | | | | |
| Include | | | worker\_thread.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | job | | job | | shared\_ptr<Job> | N/A | I | | ・job  ・valid: not NULL  ・invalid: NULL | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | Update status of job in runtime.  Executes the job and measures its time process. | | | | | | | | | | |

### isBusy

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Check job status | | isBusy() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when checking status of job. | | | | | | | | | |
| Include | | | worker\_thread.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Return value | | N/A | | Return value | | bool | N/A | O | | ・Return value  ・Valid: true, false | | |
| Processing content | | Return true when worker thread is executing a job (which is submitted before by), otherwise return false. | | | | | | | | | | |

### getLastProcessingTime

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Get last processing time | | getLastProcessingTime() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when getting last processing time of one job. | | | | | | | | | |
| Include | | | worker\_thread.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Return value | | Processing time | | Return value | | long long | 8 | O | | ・Processing time  ・Valid: [0 – 9,223,372,036,854,775,807]  ・Invalid: out of valid range | | |
| Processing content | | Return time processing (in microsecond) of the job which worker thread just executed. | | | | | | | | | | |

### process

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Worker thread processing | | process() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when starting worker thread. | | | | | | | | | |
| Include | | | worker\_thread.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | Always check message thread in the queue.  . message thread is MSG\_POST\_USER\_DATA: execute a job which is indicated by message data.  . message thread is MSG\_TIMER: execute all jobs in the internal periodic job list.  . message thread is MSG\_EXIT\_THREAD: waiting until thread completes its execution and return. | | | | | | | | | | |

## SaveCaptureImage Job

### readAndSaveImage

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Read from subscriber and save image | | readAndSaveImage() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when reading image from subscriber (Pi). | | | | | | | | | |
| Include | | | ephlib\_job.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | Read image from subscriber (Pi) via DDS network and call saveImage function.  Returns without doing anything if cannot read data from DDS network. | | | | | | | | | | |

### buildFilePathName

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | | Call name | | | Function attributes | | Created date | Author |
| Build file path name | | | buildFilePathName() | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when creating path to save image. | | | | | | | | | |
| Include | | | ephlib\_job.h | | | | | | | | | |
|  | | Variable name | | Name | Data type | | sizeﾞ | I/O | | Contents | | |
| Argument | | buffer | | File name | char\* | | 8 | O | | ・file name  ・valid: not NULL  ・invalid: NULL | | |
|  | | path | | Path directory | const char\* | | 8 | I | | ・path directory  ・valid: not NULL  ・invalid: NULL | | |
|  | | request\_DeviceId | | Device ID capture image | const char\* | | 8 | I | | ・Device ID capture image  ・valid: not NULL  ・invalid: NULL | | |
|  | | request\_CameraId | | Camera ID capture image | const char\* | | 8 | I | | ・Camera ID capture image  ・valid: not NULL  ・invalid: NULL | | |
|  | | request\_Date | | Date request | const char\* | | 8 | I | | ・Date request  ・valid: not NULL  ・invalid: NULL | | |
|  | | request\_SeqNo | | Sequence number request | unsigned int | | 4 | I | | ・Sequence number request  ・valid: [0 – 2147483647]  ・invalid: out of valid range | | |
|  | | Pricereduction\_DeviceId | | Device ID pricereduction | const char\* | | 8 | I | | ・Device ID pricereduction  ・valid: not NULL  ・invalid: NULL | | |
| Return value | | N/A | | Return value | int | | 4 | O | | ・Return value  ・valid: the total number of characters written is returned  ・invalid: a negative number | | |
| Processing content | | File path created by path, request\_DeviceId, request\_CameraId, request\_Date, request\_SeqNo, Pricereduction\_DeviceId.  If successful, the total number of characters written is returned excluding the null-character appended at the end of the string, otherwise a negative number is returned in case of failure. | | | | | | | | | | |

### saveImage

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Save image | | saveImage() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when receiving image from EPH and save image on RaspberryPi. | | | | | | | | | |
| Include | | | ephlib\_job.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | data | | Data image take topic | | Image\_Take\_Topic\_Data\* | 8 | I | | ・Data image take topic  ・valid: not NULL  ・invalid: NULL | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | Opens file path created to writes content image.  If successful, image will be saved to created file path. Otherwise, it returns without doing anything. | | | | | | | | | | |

## CaptureImage Job

### catureImage

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Capture image | | captureImage() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when EPH capture image. | | | | | | | | | |
| Include | | | ephlib\_job.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | buffer | | Image data | | vector<uchar> | N/A | O | | ・Image data  ・valid: not NULL  ・invalid: NULL | | |
| Return value | | N/A | | Return value | | bool | N/A | O | | ・Return value  ・valid: true  ・invalid: false | | |
| Processing content | | Opens camera/webcam to take image and store to buffer data.  Returns true if capture image successful, otherwise returns false. | | | | | | | | | | |

### convertToJPEG

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Convert to JPEG | | convertToJPEG() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when encoding image captured to buffer data with JPEG quality. | | | | | | | | | |
| Include | | | ephlib\_job.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | frame | | Frame data | | Mat& | N/A | I | | ・Frame data  ・valid: not NULL  ・invalid: NULL | | |
|  | | buffer | | Image data | | vector<uchar>& | N/A | O | | ・Image data  ・valid: not NULL  ・invalid: NULL | | |
| Return value | | N/A | | Return value | | bool | N/A | O | | ・Return value  ・valid: true  ・invalid: false | | |
| Processing content | | Encode frame data captured to buffer data with JPEG highest quality.  Return true if encode image successful, otherwise returns false. | | | | | | | | | | |

### publishImage

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Publish image | | publishImage() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when sending image data to DDS network after EPH captured image. | | | | | | | | | |
| Include | | | ephlib\_job.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | data | | Publish data image | | const char\* | 8 | I | | ・Publish data image  ・valid: not NULL  ・invalid: NULL | | |
|  | | length | | Length publish data image | | int | 4 | I | | ・Length publish data image  ・valid: [0 – 2147483647]  ・invalid: out of valid range | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | Send image data to DDS network.  If cannot send image data it returns without doing anything. | | | | | | | | | | |

### allocPublishData

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Allocate memory image data | | allocPublishData() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when allocating memory image data to write to DDS network to send an image. | | | | | | | | | |
| Include | | | ephlib\_job.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | length | | Length publish data image | | int | 4 | I | | ・Length publish data image  ・valid: [0 – 2147483647]  ・invalid: out of valid range | | |
| Return value | | N/A | | Return Value | | Image\_Send\_Topic\_Data\* | 8 | O | | ・Return Value  ・Valid: not NULL  ・Invalid: NULL | | |
| Processing content | | Allocates memory image data.  If cannot allocates it returns without doing anything. | | | | | | | | | | |

## RecognizeImage Job

### saveOptions

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Save image option | | saveOptions() | | | | Library | | 09/09 | QuanPDA |
| Outline of processing | | This function is called when saving option image use to recognize. | | | | | | | | | |
| Include | | ephlib\_job.h | | | | | | | | | |
|  | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | Image option | | imgOptions | | const ImageOption\* | 8 | I | | ・Image option  ・valid: not NULL  ・invalid: NULL | | |
| Return value | N/A | | Return Value | | bool | N/A | O | | ・Register for a recognition result success: true  ・Register for a recognition result failed: false | | |
| Processing content | Uses default option image if imgOptions is NULL.  Otherwise use option which user provided. | | | | | | | | | | |

### handleRequestQueue

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | Function name | | Call name | | | Function attributes | | Created date | Author |
| Handle request queue | | handleRequestQueue() | | | Library | | 09/09 | QuanPDA |
| Outline of processing | | This function is called when handling send recognition response timeout message. | | | | | | | | |
| Include | | ephlib\_job.h | | | | | | | | |
|  | Variable name | | Name | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | Queue request | | queue | vector<shared\_ptr<DnnRequest>> | N/A | I | | ・Queue request  ・valid: not NULL  ・invalid: NULL | | |
| Return value | N/A | | N/A | N/A | N/A | N/A | | N/A | | |
| Processing content | Checks if a timeout occurs with each request in queue.  If any request occurs timeout, send response message and remove it from queue.  ・Returns without doing anything if request queue is empty.  ・Returns -1 if cannot create response message.  ・Returns -2 if cannot send response message. | | | | | | | | | |

### recognizeImage

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Recognize image | | recognizeImage() | | | | Library | | 09/09 | QuanPDA |
| Outline of processing | | This function is called with each request EPH received request recognition image from DDS network. | | | | | | | | | |
| Include | | ephlib\_job.h | | | | | | | | | |
|  | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | data | | Request Data | | const Image\_Send\_Toic\_Data\* | 8 | I | | ・Request Data  ・valid: not NULL  ・invalid: NULL | | |
| Return value | N/A | | Return Value | | bool | N/A | O | | ・true: recognize successful  ・false: recognize failed | | |
| Processing content | Create and push DnnRequest to the requesting queue.  Create a callback.  Execute queueDnnJob to query a recognition result from DNN with DnnRequest, image buffer, image buffer length and callback as input arguments.  If cannot recognizes image it false. | | | | | | | | | | |

### createResultData

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | Function name | | Call name | | | Function attributes | | Created date | Author |
| Create result data | | createResultData() | | | Library | | 09/09 | QuanPDA |
| Outline of processing | | This function creates response result for EPH to send recognition result to Raspberry Pi. | | | | | | | | |
| Include | | ephlib\_job.h | | | | | | | | |
|  | Variable name | | Name | | Data type | sizeﾞ | I/O | Contents | | |
| Argument | request | | Dnn request | | DNNRequest& | 144 | I | ・Dnn request  ・Valid: not NULL  ・Invalid: NULL | | |
|  | result | | Recognition result | | string | 32 | I | ・Recognition result  ・Valid: not NULL  ・Invalid: NULL | | |
| Return value | N/A | | Response data | | Result\_Send\_Topic\_Data\* | 8 | O | ・Response Data  ・Valid: not NULL  ・Invalid: NULL | | |
| Processing content | Allocates memory for response data, copy response data to result data ad return response result.  If cannot allocate it returns without doing anything. | | | | | | | | | |

### onImageRecognized

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | Function attributes | | Created date | Author |
| On image recognize | | onImageRecognized() | | | Library | | 09/10 | BinhNK |
| Outline of processing | | | This function is called when occurs callback handler. | | | | | | | | |
| Include | | | ephlib\_job.h | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | Contents | | |
| Argument | | request | | Dnn request | | void\* | 8 | I | ・Dnn request  ・valid: not NULL  ・invalid: NULL | | |
|  | | result | | Recognition result | | string | 32 | I | ・Recognition result  ・valid: not NULL  ・invalid: NULL | | |
| Return value | | N/A | | Return value | | int | 4 | O | ・Return value  ・valid: 0  ・invalid: -1, -2 | | |
| Processing content | | Find request when callback function is called and send response recognition result.  Remove the request after done.  ・Returns without doing anything if request queue is empty.  ・Returns -1 if cannot create response message.  ・Returns -2 if cannot send response message. | | | | | | | | | |

### sendRecognitionResponse

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | Function attributes | | Created date | Author |
| Send recognition response | | sendRecognitionResponse() | | | Library | | 09/10 | BinhNK |
| Outline of processing | | | This function is called when EPH send a response to Raspberry Pi (result recognition or timeout message). | | | | | | | | |
| Include | | | ephlib\_job.h | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | Contents | | |
| Argument | | requestData | | Dnn request | | void\* | 8 | I | ・Dnn request  ・valid: not NULL  ・invalid: NULL | | |
|  | | result | | Recognition result | | string | 32 | I | ・Recognition result  ・valid: not NULL  ・invalid: NULL | | |
| Return value | | N/A | | Return value | | int | 4 | O | ・Return value  ・valid: 0  ・invalid: -1, -2 | | |
| Processing content | | Creates response data write response data to DDS network.  . If cannot create response data return -1.  . If cannot write to DDS network return -2. | | | | | | | | | |

## GetImageRecognizeResult Job

### readImage

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Read image | | readImage() | | | | Library | | 10/09 | ThaiNH |
| Outline of processing | | This function is called when reading an image from a predefined file path into a buffer. | | | | | | | | | |
| Include | | ephlib\_job.h | | | | | | | | | |
|  | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | buffer | | Buffer | | vector<uchar> | 8 | I | | ・Buffer | | |
| Return value | N/A | | Return Value | | bool | 1 | O | | ・Read image result  ・Valid: true  ・Invalid: false | | |
| Processing content | Read image from predefined file path and store into buffer.  If cannot read image it returns false. | | | | | | | | | | |

### receiveResult

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | | Call name | | | Function attributes | | | | Created date | Author |
| Receive recognition result | | | receiveResult() | | | Library | | | | 10/09 | ThaiNH |
| Outline of processing | | | This function is periodic called when Raspberry Pi’s waiting for recognition result from EPH’s response. | | | | | | | | | | | |
| Include | | |  | | ephlib\_job.h | | | | | | | | | |
|  | Variable name | | | Name | | | Data type | sizeﾞ | |  | I/O | Contents | | |
| Argument | N/A | | | N/A | | | N/A | N/A | |  | N/A | N/A | | |
| Return value | Recognition result | | | Return Value | | | string | 32 | |  | O | ・Recognition result  ・Valid: not NULL  ・Invalid: NULL | | |
| Processing content |  | Read data from DDS network.  If any valid data in DDS net work:  . Allow this device receive other images from other device.  . Receive the result string which requested by this device.  Returns NULL if has not receive any string. | | | | | | | | | | | | | |
|  |  |

### publishRecognizeImage

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | Function attributes | | | | Created date | Author |
| Publish recognize image | | publishRecognizeImage() | | | Library | | | | 10/09 | ThaiNH |
| Outline of processing | | | This function is call when publishing an image buffer over DDS network to request recognition result. | | | | | | | | | | |
| Include | | |  | ephlib\_job.h | | | | | | | | | |
|  | Variable name | | Name | | | Data type | sizeﾞ | |  | I/O | Contents | | |
| Argument | data | | Image buffer data | | | const char\* | 8 | |  | I | ・Image buffer data  ・Valid: not NULL  ・Invalid: NULL | | |
|  | length | | Image buffer length | | | int | 4 | |  | I | ・Image buffer length  ・Valid: [0 – 2147483647]  ・Invalid: out of valid range | | |
| Return value | N/A | | N/A | | | N/A | N/A | |  | N/A | N/A | | |
| Processing content |  | Creates publish data copy image buffer to publish data.  Writes publish data to DDS network.  If cannot publish image to DDS network it returns without doing anything. | | | | | | | | | | | | |
|  |

### allocPublishData

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | | Call name | | | | | | | | Function attributes | | | | Created date | | 作成者 | |
| Allocate publish data | | | allocPublishData() | | | | | | | | Library | | | | 10/09 | | ThaiNH | |
| Outline of processing | | | This function is called when allocating memory image data to write to DDS network to request recognition result for an image. | | | | | | | | | | | | | | | | | | |
| Include | | | ephlib\_job.h | | | | | | | | | | | | | | | | | | |
|  | Variable name | | | Name | | | | Data type | | sizeﾞ | | I/O | | | | Contents | | | | | |
| Argument | length | | | Image buffer length | | | | int | | 4 | | I | | | | ・Image buffer length  ・Valid: [0 – 2147483647]  ・Invalid: out of valid range | | | | | |
| Return value | N/A | | | Return Value | | | | Image\_Send\_Topic\_Data\* | | 8 | | O | | | | ・Return Value  ・Valid: not NULL  ・Invalid: NULL | | | | | |
| Processing content | Allocates and returns memory image data. | | | | | | | | | | | | | | | | | | | | |
|  |  |  | |  |  | |  | |  | |  | |  | |  | |  | |  | |  | |

## DnnRecognition

### queueDnnJob

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Queue Dnn Job | | queueDnnJob() | | | | Library | | 26/9 | HoaDV |
| Outline of processing | | | This function is called when executing ddn job. | | | | | | | | | |
| Include | | | ephlib\_dnnjob.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | request | | Request recognition | | void\* | 8 | I | | ・Request recognition  ・Valid: not NULL  ・Invalid: NULL | | |
| Return value | | N/A | | Return value | | bool | N/A | O | | ・Return value  ・Valid: true  ・Invalid: false | | |
| Processing content | | Finds first free buffer and writes request recognition.  Returns false if:  . Dnn thread is terminated.  . Size recognition image too big.  . Cannot found free buffer. | | | | | | | | | | |

### start

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Start Dnn recognition thread | | start() | | | | Library | | 26/9 | HoaDV |
| Outline of processing | | | This function is called when starting DNN recognition thread | | | | | | | | | |
| Include | | | ephlib\_dnnjob.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | Creates and starts a thread to execute DNN request. If the thread has started before, returns without doing anything. | | | | | | | | | | |

### join

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Join Dnn recognition thread | | join() | | | | Library | | 26/9 | HoaDV |
| Outline of processing | | | This function is called when waiting until thread completes its execution | | | | | | | | | |
| Include | | | ephlib\_dnnjob.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | If DNN has started, blocks caller thread until DNN thread finishes its execution.  Otherwise, returns without doing anything. | | | | | | | | | | |

### stop

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Stop Dnn recognition thread | | stop() | | | | Library | | 26/9 | HoaDV |
| Outline of processing | | | This function is called when stoping DNN recognition thread | | | | | | | | | |
| Include | | | ephlib\_dnnjob.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | If DNN thread has started, creates an exit message, and notifies DNN thread.  Otherwise, returns without doing anything. | | | | | | | | | | |

### stopAndWait

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Stop and wait Dnn recognition thread | | stopAndWait() | | | | Library | | 26/9 | HoaDV |
| Outline of processing | | | This function is called when stopping all job in worker thread. | | | | | | | | | |
| Include | | | ephlib\_dnnjob.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | Combines behavior stop() and join(). | | | | | | | | | | |

### readReponse

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Read response pipe | | readReponse() | | | | Library | | 26/9 | HoaDV |
| Outline of processing | | | This function is called when reading response pipe. | | | | | | | | | |
| Include | | | ephlib\_dnnjob.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | result | | Result string | | string | 32 | I | | ・Result string  ・Valid: not NULL  ・Invalid: NULL | | |
| Return value | | N/A | | Return value | | int | 4 | O | | ・Return value  ・Valid: 0  ・Invalid: -1 | | |
| Processing content | | Reads result code & result string from response pipe of DNN application (provided by TTEC).  Returns without doing anything if:  . Cannot open response pipe.  . Cannot read result code.  . Cannot read result string. | | | | | | | | | | |

### writeRequest

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Write request pipe | | writeRequest() | | | | Library | | 26/9 | HoaDV |
| Outline of processing | | | This function is called when reading response pipe. | | | | | | | | | |
| Include | | | ephlib\_dnnjob.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | img | | Recognition image | | const char\* | 8 | I | | ・Result string  ・Valid: not NULL  ・Invalid: NULL | | |
|  | | len | | Length recognition image | | int | 4 | I | | ・Length recognition image  ・Valid: [0 – 2147483647]  ・Invalid: out of valid range | | |
| Return value | | N/A | | Return value | | int | 4 | O | | ・Return value  ・Valid: 0  ・Invalid: -1 | | |
| Processing content | | Writes image to request pipe of DNN application (provided by TTEC).  Returns -1 if:  . Cannot open request pipe.  . Cannot write recognition to request pipe. | | | | | | | | | | |

### dequeueBuffer

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Dequeue buffer | | dequeueBuffer() | | | | Library | | 26/9 | HoaDV |
| Outline of processing | | | This function is called when finding first free buffer. | | | | | | | | | |
| Include | | | ephlib\_dnnjob.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | reading | | Read option | | bool | N/A | I | | ・Read option  ・Valid: true, false | | |
| Return value | | N/A | | Return value | | int | 4 | O | | ・Return value  ・Valid: >= 0  ・Invalid: < 0 | | |
| Processing content | | Dequeues a buffer that selected by read option. If read option is true, dequeues Reading buffer. Otherwise dequeues Writing Buffer.  If no buffer element is available, waits till at least 1 element is available in target buffer.  Returns:  . buffer index when dequeue successfully.  . -1 when DNN thread is requested to be terminated. | | | | | | | | | | |

### callCallbackWithResult

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Callback with result | | callCallbackWithResult() | | | | Library | | 26/9 | HoaDV |
| Outline of processing | | | This function is called when sending callback to raspberry Pi. | | | | | | | | | |
| Include | | | ephlib\_dnnjob.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | hdr | | DDS receive structure data | | DDS\_RECEIVE\_HDR& | N/A | I | | ・DDS receive structure data  ・Valid: not NULL  ・Invalid: NULL | | |
| Return value | | result | | Result string | | string | 32 | I | | ・Result string  ・Valid: not NULL  ・Invalid: NULL | | |
| Processing content | | If callback is registered, calls that callback function to notify about recognition result. | | | | | | | | | | |

### process

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Dnn processing | | process() | | | | Library | | 26/9 | HoaDV |
| Outline of processing | | | This function is called when starting DNN thread. | | | | | | | | | |
| Include | | | ephlib\_dnnjob.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | Dequeues reading buffer to get DNN recognition request, recognize image. After recognizing, read result and notify by callback.  Above processing is terminated when it receives an exit request by stop(). | | | | | | | | | | |

## ephlib

### SaveCapturedImage

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Save capture image | | SaveCapturedImage() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when saving image periodically. | | | | | | | | | |
| Include | | | ephlib.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | Topic name | | topicName | | const char\* | 8 | I | | ・Topic name  ・Valid: not NULL  ・Invalid: NULL | | |
|  | | Folder path | | folderPath | | const char\* | 8 | I | | ・Folder path  ・Valid: not NULL  ・Invalid: NULL | | |
| Return value | | Job ID | | Return value | | CMULONG | 8 | O | | ・Job ID  ・Valid: not NULL  ・Invalid: NULL | | |
| Processing content | | Creates and starts a worker thread if thread has not started.  Set participant for DDS network.  Creates new job save image and add to internal periodic list.  . Returns job ID if add job successful.  . Returns NULL is cannot create job. | | | | | | | | | | |

### GetImageRecognizeResult

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Get image recognition result | | GetImageRecognizeResult() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when getting recognition result periodically. | | | | | | | | | |
| Include | | | ephlib.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | Request topic name | | requestTopicName | | const char\* | 8 | I | | ・Request topic name  ・Valid: not NULL  ・Invalid: NULL | | |
|  | | Response topic name | | responseTopicName | | const char\* | 8 | I | | ・Response topic name  ・Valid: not NULL  ・Invalid: NULL | | |
|  | | File path | | filePath | | const char\* | 8 | I | | ・File path  ・Valid: not NULL  ・Invalid: NULL | | |
|  | | Call back | | callback | | RecognitionCallback | N/A | I | | ・Call back  ・Valid: not NULL  ・Invalid: NULL | | |
| Return value | | Job ID | | Return value | | CMULONG | 8 | O | | ・Job ID  ・Valid: not NULL  ・Invalid: NULL | | |
| Processing content | | Creates and starts a worker thread if thread has not started.  Sets participant for DDS network.  Creates new job get recognition result and add to internal periodic list.  . Returns job ID if add job successful.  . Returns NULL is cannot create job.  . callback is NULL | | | | | | | | | | |

### RecognizeImage

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Recognize image | | RecognizeImage() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when getting recognition result periodically. | | | | | | | | | |
| Include | | | ephlib.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | Request topic name | | requestTopic | | const char\* | 8 | I | | ・Request topic name  ・Valid: not NULL  ・Invalid: NULL | | |
|  | | Response topic name | | answerTopic | | const char\* | 8 | I | | ・Response topic name  ・Valid: not NULL  ・Invalid: NULL | | |
|  | | Image option | | option | | const ImageOption\* | 8 | I | | ・Image option  ・Valid: not NULL  ・Invalid: NULL | | |
| Return value | | Job ID | | Return value | | CMULONG | 8 | O | | ・Job ID  ・Valid: not NULL  ・Invalid: NULL | | |
| Processing content | | Creates and starts a worker thread if thread has not started.  Set participant for DDS network.  Creates new job get recognition result and add to internal periodic list.  . Returns job ID if add job successful.  . Returns NULL is cannot create job.  . option is NULL. | | | | | | | | | | |

### CaptureImage

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Recognize image | | RecognizeImage() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when getting recognition result periodically. | | | | | | | | | |
| Include | | | ephlib.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | Topic name | | topicName | | const char\* | 8 | I | | ・Topic name  ・Valid: not NULL  ・Invalid: NULL | | |
| Return value | | Job ID | | Return value | | CMULONG | 8 | O | | ・Job ID  ・Valid: not NULL  ・Invalid: NULL | | |
| Processing content | | Creates and starts a worker thread if thread has not started.  Sets participant for DDS network.  Creates new job get recognition result and add to internal periodic list.  . Returns job ID if add job successful.  . Returns NULL is cannot create job. | | | | | | | | | | |

### WaitAllJob

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Wait all job | | WaitAllJob () | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when waiting until all thread completes its execution. | | | | | | | | | |
| Include | | | ephlib.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | If worker thread has started, then blocks current caller thread and waits until worker thread finishes its execution.  Otherwise, it returns without doing anything. | | | | | | | | | | |

### StopJob

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Stop job | | WaitAllJob() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when removing periodically thread. | | | | | | | | | |
| Include | | | ephlib.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | Job ID | | jobId | | CMULONG | 8 | I | | ・Job ID  ・Valid: not NULL  ・Invalid: NULL | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | If worker thread has started, it returns without doing anything.  Call functions removePeriodicJob. | | | | | | | | | | |

### ShutdownAllJob

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Item Number | １ | | Function name | | Call name | | | | Function attributes | | Created date | Author |
| Shutdown all job | | ShutdownAllJob() | | | | Library | | 27/8 | BinhNK |
| Outline of processing | | | This function is called when stopping both worker thread and Dnn thread. | | | | | | | | | |
| Include | | | ephlib.h | | | | | | | | | |
|  | | Variable name | | Name | | Data type | sizeﾞ | I/O | | Contents | | |
| Argument | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Return value | | N/A | | N/A | | N/A | N/A | N/A | | N/A | | |
| Processing content | | If worker thread has started, waits and stops worker thread.  If Dnn thread has started, waits and stops Dnn thread.  Shutdown participant DDS network.  If worker thread is not started, it returns without doing anything. | | | | | | | | | | |

Template's Revision History

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Rev. No.  (X.YY) | Date (YYYY-MM-DD) | Section No. Changed | Change Description | Author | Reviewed by | Approved by |
| 1.01 | 2011-04-26 | 6 | Delete traceability matrix part | GiangNT | ThuyTTP  TrinhNTT  Kojima-san |  |
| 2.00 | 2011-07-28 |  | Approved by managers | GiangNT | GD.Tabe  PM.AnhPT | GD.Tabe  PM.AnhPT |
| 2.01 | 2012-05-08 | Revision History | Fix date format | ToanDT |  |  |
| 3.00 | 2012-06-06 | Template's RH | Approved by manager | ToanDT |  | AnhPT |
| 3.01 | 2018-11-06 | All | Change document into new template/logo | HienNN | [QAM] T.Ninomiya | [GD] K.Shigenaka  [PM] TrungDV, ThinhND |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |