

FPT UNIVERSITY FLYING OBJECT

DETAIL DESIGN DOCUMENT

Class design

**Project Code:**

**Document Code: FUFO-DD – v.1.0.1**

**Ha Noi, August 08, 2012**

Record of change

\*A - Added M - Modified D – Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Effective Date | Changed Items | A\* M, D | Change Description | New Version |
| 08/08/2012 | Create new document | A |  | 1.0.1 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

SIGNATURE PAGE

**ORIGINATOR:** NguyenLK 08/ 08/ 2012

Developer

**REVIEWERS:** <Name> <Date>

<Position>

<Name, if it’s needed> <Date>

<Position>

**APPROVAL:** <Name> <Date>

<Position>

TABLE OF CONTENTS

1 Introduction 6

1.1 Purpose 6

1.2 Definitions, Acronyms and Abbreviations 6

1.3 References 7

1.4 Overview 7

2 COMMON package and mechanism 8

2.1 Application on Computer 8

2.1.1 Class diagram 8

2.1.2 CommandControl class 10

2.1.3 Control class 12

2.1.4 AOC class 13

2.1.5 InstrumentControl class 17

2.1.6 StatusControl class 18

2.1.7 StatusMessage class 21

2.1.8 VideoControl class 22

2.2 Application on Phone 25

2.2.1 Class diagram 25

2.2.2 CommandControl class 27

2.3 Error, exception handling 29

2.3.1 Class Diagram 29

2.3.2 Usage mechanism 29

2.4 Log, trace and debug 29

2.5 Performance optimizing mechanism 29

2.6 Multilingual processing 29

3 Diagrams 30

4 Packages 32

4.1 xxx Package 32

4.1.1 Class diagram 32

4.1.2 External interface 32

4.1.3 XXX class 32

4.2 ...... 33

5 Other considerations 34

6 Appendix 35

# Introduction

## Purpose

This part will give general description of the document including:

* Purpose of document.
* What are contained in the doc
* Reader of document
* Other description about scope of document, limitation,...

Ex.

XXX class design document contains descriptions of all classes of the YYYY system including:

Class diagrams that describe the static relation of all classes in the system

Dynamic diagrams such as collaboration diagram, sequence diagram, activities diagram

Description of class in detail

Common mechanism in implementing is also defined in this document

Developer and tester will base on this design to implement classes and unit test.

## Definitions, Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
| Abbreviations | Description | Comment |
| TBD | To be decided | It means “not decided yet” |
| Windows DDK | Windows Device Development Kit | Development Kit from Microsoft to develop device driver for Windows 2000/XP/Server 2003 OS |
| ESC/P | Escape Printing command | A page description language used on Epson ink jet printers |
|  |  |  |
|  |  |  |

## References

List all the reference document such as: other document of the system, or the technical article,...

|  |  |
| --- | --- |
| Document Number | Title |
| 01 | Data design document |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

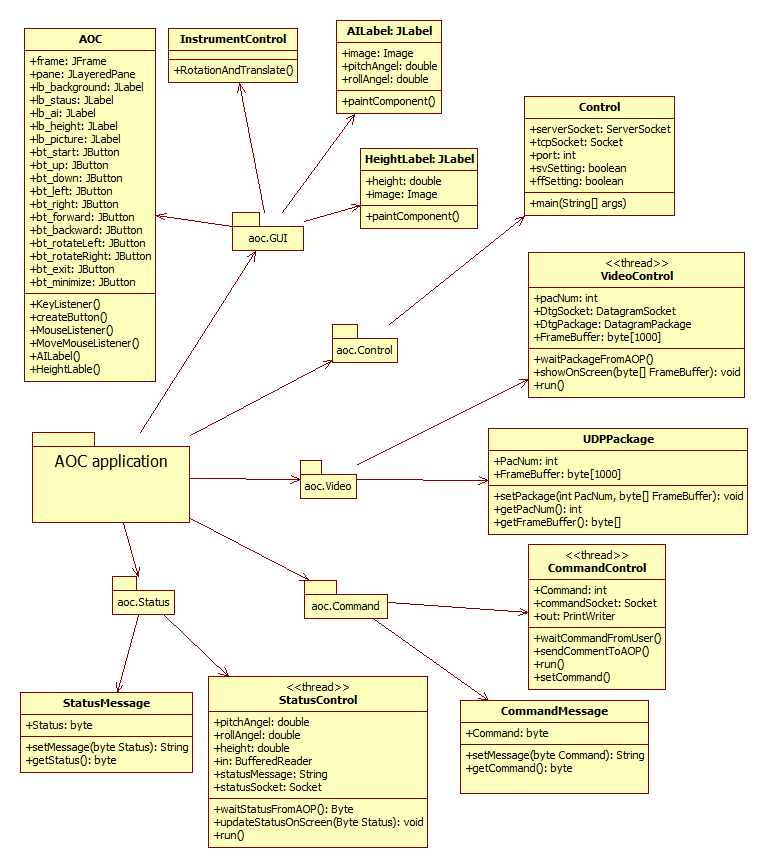
## Overview

General overview of the detail design such as what is the structure of the document

# COMMON package and mechanism

## Application on Computer

### Class diagram



|  |  |  |
| --- | --- | --- |
| No | Class Name | Description |
| 01 | CommandControl | This class uses to receive pressing keyboard from users and send it to Android phone via TCP. |
| 02 | CommandMessage | This class includes “set” and “get” method to establish command message structure. |
| 03 | Control | This class is main class uses to control all of threads and GUI in application on computer. |
| 04 | AOC | This class uses to create new Graphic User Interface for Application On Computer. It also Key Listener and Mouse Listener to monitor behavior of user |
| 05 | AILabel | Uses to draw attitude indicator |
| 06 | HeightLabel | Uses to draw height indicator |
| 07 | InstrumentControl | This class includes method to draw height indicator, attitude indicator on Graphic User Interface. |
| 08 | StatusControl | This class uses to receive status from AOP and update it to screen. |
| 09 | StatusMessage | This class includes “set” and “get” method to establish status message structure. |
| 10 | VideoControl | This class uses to receive video frame from AOP and update it to screen. |
| 12 | UDPPackage | This class includes “set” and “get” method to establish UDP package structure. |

### CommandControl class

This class extends Thread uses to receive command when user press keyboard and send it to Android phone via TCP.

#### Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Attribute | Type | Default | Note | Description |
| 01 | commandSocket | Socket | Null |  | TCP socket user to send command to AOP |
| 02 | out | PrintWriter | Null |  | Uses OutputStream to send command to AOP |
| 03 | command | Int | 0 |  | Command to send to AOP |
| 04 | county | Int | 0 |  | Use to check time to send y signal to AOP |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | CommandControl | Default constructor |
| 02 | CommandControl | Constructor uses to input TCP Socket |
| 03 | setCommand | Set command for CommandControl class |
| 04 | setCommandSocket | Set commandSocket for CommandControl class |
| 05 | run | This method will run when thread start. |
| 06 | waitCommandFromUser | Uses to wait command from user input |
| 07 | sendCommandToAOP | Uses to send command to AOP |

* + - 1. CommandControl Method

This method is the constructor uses to input the command socket

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | commandSocket | Socket | In |  | This is TCP Socket |

* + - 1. setCommand Method

This method is the constructor uses to input the command socket

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | command | int | In |  | This is command to control FUFO |

* + - 1. setCommandSocket Method

This method is the constructor uses to input the command socket

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | commandSocket | Socket | In |  | This is TCP Socket |

* + - 1. run Method

This method is the default method on the thread. Codes will be executed when thread is started.

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | command | int | In |  | This is command to control FUFO |

##### Implementation

Initiate PrintWriter to send command to Android phone by input two parameters: OutputStream of commandsocket and Boolean variable “true”.

Create a while loop repeat forever.

In while loop: if command differs with zero, send command to AOP, else send “y” signal to AOP for AOP know that connection is good. Time between two transmittals is 500 milliseconds.

### Control class

This class is main class uses to control all of threads and GUI in application on computer.

#### Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Attribute | Type | Default | Note | Description |
| 01 | serverSocket | ServerSocket | Null | private/static | This is Server Socket on AOC |
| 02 | tcpSocket | Socket | Null | private/static | This is Socket to connect with AOP |
| 03 | PORT | int | 8888 | final/static | Port of server |
| 04 | stct | StatusControl | Null | Public/static | Thread status control |
| 05 | cmct | CommandControl | Null | Public/static | Thread command control |
| 06 | vdct | VideoControl | Null | public/static | Thread video control |
| 07 | svSetting | boolean | false | public/static | Flag to check socket between AOP and AOC |
| 08 | ffSetting | boolean | false | public/static | Flag to check socket between AOP and FUFO |
| 09 | start | int | 0 | public/static | Flag to check socket whether start button pressed |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | main | main method of the system |

* + - 1. main Method

This method is main method of the system

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | args | String Array | In |  | Arguments input |

##### Implementation

Initiate GUI AOC.

Start server

Wait for client connecting

Initiate and start three threads

### AOC class

This is GUI class.

#### Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Attribute | Type | Default | Note | Description |
| 01 | frame | JFrame | null |  | Main Frame of application |
| 02 | pane | JLayeredPane | null |  | Pane contains all component of GUI |
| 03 | lb\_background | JLabel | null |  | Set image background by this label |
| 04 | lb\_circle | JLabel | null |  | Circle of attitude indicator |
| 05 | lb\_status | JLabel | null |  | This label checks connection status of system |
| 06 | lb\_picture | JLabel | null |  | Video frame streamed from AOP |
| 07 | lb\_setPoint | JLabel | null |  | Height that user want |
| 08 | lb\_text | JLabel | null |  | Text in label status |
| 09 | lb\_ai | AILabel | null |  | Attitude indicator |
| 10 | lb\_height | HeightLabel | null |  | Height indicator |
| 12 | bt\_start | JButton | null |  | Button start |
| 13 | bt\_right | JButton | null |  | Button right |
| 14 | bt\_left | JButton | null |  | Button left |
| 15 | bt\_forward | JButton | null |  | Button forward |
| 16 | bt\_backward | JButton | null |  | Button backward |
| 17 | bt\_up | JButton | null |  | Button up |
| 18 | bt\_down | JButton | null |  | Button down |
| 19 | bt\_rotateLeft | JButton | null |  | Button rotate left |
| 20 | bt\_rotateRight | JButton | null |  | Button rotate right |
| 21 | bt\_exit | JButton | null |  | Button exit |
| 22 | bt\_minimize | JButton | null |  | Button minimize |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | AOC | constructor |
| 02 | createButton | create a button in frame base on position, path of icon button |
| 03 | mouseListener | Mouse listener |
| 04 | keyListener | Key listener |
| 05 | moveMouseListener | Mouse move listener |
| 06 | AILabel | Label for attitude indicator |
| 07 | HeightLabel | Lable for height indicator |

* + - 1. AOC Constructor

This method is the AOC constructor.

##### Implementation

Initiate new JFrame for system. Set size map with size in screen design.

Set hide title bar, transparent, not resize.

Add components to frame base on position in screen design.

Add moveMouseListener, mouseListener, keyListener for this frame.

* + - 1. createButton Method

This method is used to create a button in frame base on position, path of icon button

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | text | String | In |  | Name of button |
| 02 | path | String | In |  | Path of icon button |
| 03 | x | int | In |  | the x-coordinate of button |
| 04 | y | int | In |  | the y-coordinate of button |
| 05 | width | int | In |  | Width of button |
| 06 | height | int | In |  | Height of button |

##### Implementation

SetContentAreaFilled for button is false

Set empty border.

Add button to pane with layer parameter is zero

Add mouse listener and key listener of this class

Return a button with upper parameter.

* + - 1. KeyListener

This is object uses to listen the key input from user.

##### Implementation

Set correlative command and icon on GUi when user presses keyboard.

* + - 1. mouseListener

This is object uses to listen the mouse input from user.

##### Implementation

Set correlative command and icon on GUI when user presses mouse.

* + - 1. moveMouseListener

This is object uses to listen the move mouse from user.

##### Implementation

This objects uses for user to click and move frame to anywhere they want on desktop screen.

### AILabel class

This class is extended JLabel uses to override method paintComponent to draw attitude indicator.

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | image | Image | In | null | Image for attitude indicator |
| 02 | pitchAngel | Double | In | 0 | Pitch angel of FUFO |
| 03 | rollAngel | Double | In | 0 | Roll angel of FUFO |

##### Implementation

In paintComponent medthod, use RotateAndTranslate method of InstrumentControl to draw attitude indicator base on pitch angel and roll angel.

### HeightLabel class

This class is extended JLabel uses to override method paintComponent to draw height indicator.

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | image | Image | In | null | Image for height indicator |
| 02 | pitchAngel | Double | In | 0 | Pitch angel of FUFO |
| 03 | rollAngel | Double | In | 0 | Roll angel of FUFO |

##### Implementation

In paintComponent medthod, use RotateAndTranslate method of InstrumentControl to draw attitude indicator base on height.

### InstrumentControl class

This class show methods to draw attitude indicator and height indicator on screen

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | RotateAndTranslate | Method to rote and translate image base on pitch angel and roll angel |

* + - 1. RotateAndTranslate Method

This Method to rote and translate image base on pitch angel and roll angel .

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 10 | Pe | Graphics | In |  | The paint area event where the image will be displayed |
| 02 | Img | Image | In |  | The image to display |
| 03 | alphaRot | Double | In |  | The angle of rotation in radian |
| 04 | alphaTrs | Double | In |  | The angle of translation direction in radian, expressed in the rotated image coordonate system |
| 05 | ptImg | Point | In |  | The location of the left upper corner of the image to display in the paint area in nominal situation |
| 06 | deltaPx | int | In |  | The translation distance in pixel |
| 07 | ptRot | Point | In |  | The location of the rotation point in the paint area |
| 08 | scaleFactor | **float** | In |  | Multiplication factor on the display image |

##### Implementation

Apply algorithm from <http://www.codeproject.com/Articles/27411/C-Avionic-Instrument-Controls> to implement this method.

### StatusControl class

This class extends Thread uses to receive status from Android phone via TCP.

#### Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Attribute | Type | Default | Note | Description |
| 01 | aoc | AOC | Null |  | GUI AOC |
| 02 | statusSocket | Socket | Null |  | Uses to receive status from AOP |
| 03 | in | BufferedReader | Null |  | Uses to read buffer |
| 04 | sm | StatusMessage | Null |  | Use to analyze message |
| 05 | pitchAngel | Double | 0 |  | Pitch angel update to attitude indicator |
| 06 | rollAngel | Double | 0 |  | Roll angel update to attitude indicator |
| 07 | height | Double | 0 |  | Height update to height indicator |
| 08 | statusMessage | String | Null |  | Status message |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | StatusControl | Default constructor |
| 02 | StatusControl | Constructor uses to input TCP Socket and AOC |
| 03 | waitStatusFromeAOP | Uses to wait status from AOP via TCP |
| 04 | updateStatusOnScreen | Uses to update pitch, roll angel, height to screen. |
| 05 | run | This method will run when thread start. |

* + - 1. StatusControl Method

This method is the constructor uses to input the status socket and aoc

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | statusSocket | Socket | In |  | This is TCP Socket |
| 02 | aoc | AOC | In |  | GUI |

* + - 1. waitStatusFromeAOP Method

This method uses to receive status from AOP

##### Implementation

Read line from bufferedReader to statusMessage.

Use StatusMessage class to analyze pitch angel, roll angel and height and set to variables pitchAngel, rollAngel, height.

* + - 1. updateStatusOnScreen Method

This method uses to update roll angel, pitch angel, height to screen.

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | pitchAngel | Double | In |  | Pitch Angel |
| 02 | rollAngel | Double | In |  | Roll angel |
| 03 | height | Double | In |  | heigh |

##### Implementation

Set pitchAngel and rollAngel of lb\_ai of aoc is pitchAngel and rollAngel.

Set pitchAngel of lb\_height of aoc is height.

Repaint the frame.

* + - 1. run Method

This method is the default method on the thread. Codes will be executed when thread is started.

##### Implementation

Create a while loop repeat forever.

In while loop: include two method updateStatusOnScreen and waitStatusFromeAOP.

### StatusMessage class

This class uses analyze statusMessage received from AOP

#### Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Attribute | Type | Default | Note | Description |
| 01 | pitchAngel | Double | 0 |  | Pitch angel update to attitude indicator |
| 02 | rollAngel | Double | 0 |  | Roll angel update to attitude indicator |
| 03 | height | Double | 0 |  | Height update to height indicator |
| 04 | statusMessage | String | Null |  | Status message |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | setStatusMessge | Uses to create message from pitch angel, roll angel and height. |
| 02 | getStatus | Uses to get pitch angel, roll angel and height from message. |

* + - 1. setStatusMessge Method

This method uses to create message from pitch angel, roll angel and height

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | pitchAngel | Double | In |  | Pitch angel update to attitude indicator |
| 02 | rollAngel | Double | In |  | Roll angel update to attitude indicator |
| 03 | height | Double | In |  | Height update to height indicator |
| 04 | statusMessage | String | Out |  | statusmessage |

##### Implementation

Parse to string and combine three upper parameters to a string, between two parameters is a comma.

* + - 1. getStatus Method

This method uses to receive status from AOP

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | statusMessage | String | Out |  | statusmessage |

##### Implementation

Split string by comma, convert to double and set it to three variable pitch angel, roll angel and height.

### VideoControl class

This class extends Thread uses to receive video from Android phone via UDP.

#### Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Attribute | Type | Default | Note | Description |
| 01 | aoc | AOC | Null |  | GUI AOC |
| 02 | udpSocket | DatagramSocket | Null |  | Uses to receive video from AOP |
| 03 | udpPackage | DatagramPackage | Null |  | Datagram package |
| 04 | pk | UDPpackage | Null |  |  |
| 05 | pacBuffer | Byte[] | Null |  | Buffer uses to save package |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | VideoControl | Default constructor |
| 02 | VideoControl | Constructor uses to input AOC |
| 03 | waitPackageFromeAOP | Uses to wait package from AOP via UDP |
| 04 | showOnScreen | Update image to screen. |
| 05 | run | This method will run when thread start. |

* + - 1. StatusControl Method

This method is the constructor uses to input aoc

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | aoc | AOC | In |  | GUI |

* + - 1. waitPackageFromeAOP Method

This method uses to receive package from AOP

##### Implementation

Initiate new DatagramSocket with port 4444.

Initiate new DatagramPackage with two parameters: pacBuffer and length of this buffer

Receive udpPackage by udpsocket

* + - 1. showStatusOnScreen Method

This method uses to update roll angel, pitch angel, height to screen.

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | udpPackage | DatagramPackage | In |  | Udp package receive from aop |

##### Implementation

Rescale picture with size 800,600

Show on screen.

* + - 1. run Method

This method is the default method on the thread. Codes will be executed when thread is started.

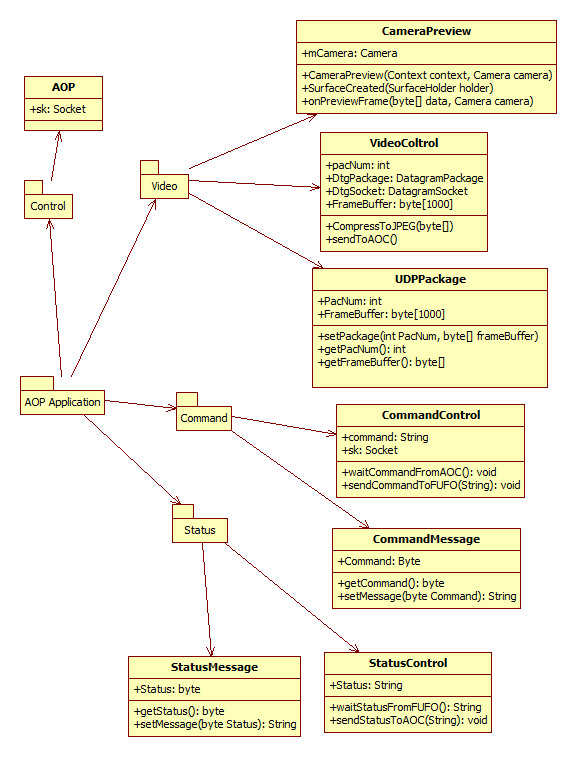
##### Implementation

Create a while loop repeat forever.

In while loop: include two method showOnScreen and waitPackageFromeAOP

## Application on Phone

### Class diagram



|  |  |  |
| --- | --- | --- |
| No | Class Name | Description |
| 01 | CommandControl | This class uses to receive pressing keyboard from users and send it to Android phone via TCP. |
| 02 | CommandMessage | This class includes “set” and “get” method to establish command message structure. |
| 03 | Control | This class is main class uses to control all of threads and GUI in application on computer. |
| 04 | AOP | This class uses to create new Graphic User Interface for Application On Computer. It also Key Listener and Mouse Listener to monitor behavior of user |
| 05 | InstrumentControl | This class includes method to draw height indicator, attitude indicator on Graphic User Interface. |
| 06 | StatusControl | This class uses to receive status from AOP and update it to screen. |
| 07 | StatusMessage | This class includes “set” and “get” method to establish status message structure. |
| 08 | VideoControl | This class uses to receive video frame from AOP and update it to screen. |
| 09 | UDPPackage | This class includes “set” and “get” method to establish UDP package structure. |

### CommandControl class

This class extends Thread uses to receive command when user press keyboard and send it to Android phone via TCP.

#### Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Attribute | Type | Default | Note | Description |
| 01 | commandSocket | Socket | Null |  | TCP socket user to send command to AOP |
| 02 | out | PrintWriter | Null |  | Uses OutputStream to send command to AOP |
| 03 | command | Int | 0 |  | Command to send to AOP |
| 04 | county | Int | 0 |  | Use to check time to send y signal to AOP |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | CommandControl | Default constructor |
| 02 | CommandControl | Constructor uses to input TCP Socket |
| 03 | setCommand | Set command for CommandControl class |
| 04 | setCommandSocket | Set commandSocket for CommandControl class |
| 05 | run | This method will run when thread start. |
| 06 | waitCommandFromUser | Uses to wait command from user input |
| 07 | sendCommandToAOP | Uses to send command to AOP |

* + - 1. CommandControl Method

This method is the constructor uses to input the command socket

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | commandSocket | Socket | In |  | This is TCP Socket |

* + - 1. setCommand Method

This method is the constructor uses to input the command socket

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | command | int | In |  | This is command to control FUFO |

* + - 1. setCommandSocket Method

This method is the constructor uses to input the command socket

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | commandSocket | Socket | In |  | This is TCP Socket |

* + - 1. run Method

This method is the default method on the thread. Codes will be executed when thread is started.

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | In/out | Default | Description |
| 01 | command | int | In |  | This is command to control FUFO |

##### Implementation

Initiate PrintWriter to send command to Android phone by input two parameters: OutputStream of commandsocket and Boolean variable “true”.

Create a while loop repeat forever.

In while loop: if command differs with zero, send command to AOP, else send “y” signal to AOP for AOP know that connection is good. Time between two transmittals is 500 milliseconds

## Error, exception handling

### Class Diagram

Describe class like in common package

### Usage mechanism

Common mechanism of exception handling

## Log, trace and debug

## Performance optimizing mechanism

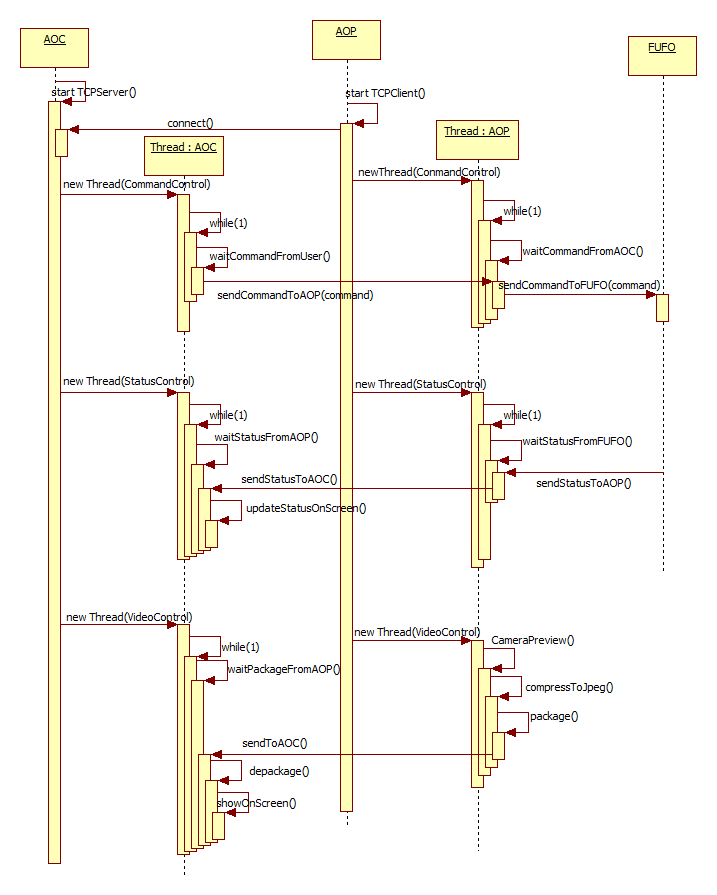
## Multilingual processing

# Diagrams

Describe diagrams in system such as collaboration diagram, sequence diagram, activities diagram and state chart for some functionalities of the system

**Example**

....



# Packages

|  |  |  |  |
| --- | --- | --- | --- |
| No | Package | Language | Description |
| 01 | <package name> | C++, Java | <brief description of package; can be one sentence tell what the method does> |
|  |  |  |  |
|  |  |  |  |

## xxx Package

### Class diagram

<Class diagram figure>

|  |  |  |
| --- | --- | --- |
| No | Class Name | Description |
| 01 | <Name of class> | <Brief description about class ex. One sentence to tell what the class is for, what does it encapsulate> |

### External interface

Describe the external interface of the package (exported classes, methods).

### XXX class

<Class description>

#### Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Attribute | Type | Default | Note | Description |
| 01 | <Attribute name> | int |  | Public/ Static | <Description of attribute> |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

#### Methods

|  |  |  |
| --- | --- | --- |
| No | Method | Description |
| 01 | <method name> | <brief description of method. can be one sentence tell what the method does> |
|  |  |  |
|  |  |  |

#### xxxx method

<Method declaration>

<method description, it must be compliance with the brief description in the upper class list>

##### Parameters & return

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Parameter | Type | in/out | Default | Description |
| 01 | <parameter name> | int |  |  | <Description of parameter, the special criteria such as boundary should be stated> |
|  |  |  |  |  |  |
|  | <return > |  |  |  |  |

##### Implementation

<How to implement the method, it can be in pseudo code or activity diagram or just words>

## ......

# Other considerations

<[This section provides a description of other design elements that were considered as alternatives in selection process for the above class design, i.e. a brief explanation of advantages and disadvantages of the selected package relationships and/or class implementation in comparison with others. It should be a clear answer to the question why the above class design is selected for this system, not the others.>

# Appendix