|  |  |  |
| --- | --- | --- |
| Department: [E Technical Department] | |  |
| Approved by | Checked by | Created by |
|  |  |  |

Becky

Software Architecture Design Specifications

Version 1

Revision history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ver No. | Created on | Revised contents  (changed parts, Changed content and impacted related documents) | Created by | Approved by |
| 01 | 2017/07/18 | Initial release | Harada | Tamura |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Index

[1. Purpose 4](#_Toc2875216)

[2. Scope of Application 4](#_Toc2875217)

[3. Definition/References 4](#_Toc2875218)

[3.1. Reference 4](#_Toc2875219)

[3.2. Definition 4](#_Toc2875220)

[4. Software Architecture Overview 5](#_Toc2875221)

[4.1. Architecture schematic diagram 5](#_Toc2875222)

[4.2. User interface (UI) 6](#_Toc2875223)

[4.3. Scanning/Measurement main 7](#_Toc2875224)

[4.4. System architecture 16](#_Toc2875225)

[4.5. Software architecture 17](#_Toc2875226)

[5. Development environment 17](#_Toc2875227)

[5.1. Main CPU board（Master/Slave） 17](#_Toc2875228)

[5.2. Sub CPU board (optical / base) 18](#_Toc2875229)

[6. Operating environment 18](#_Toc2875230)

[6.1. Main CPU board（Master/Slave） 18](#_Toc2875231)

[6.2. Sub CPU board (optical / base) 18](#_Toc2875232)

[7. Design restrictions 18](#_Toc2875233)

[8. Software item definition 19](#_Toc2875234)

[8.1. Software item list 19](#_Toc2875235)

[8.2. Software safety class classification 19](#_Toc2875236)

[8.3. Software item interface definition 19](#_Toc2875237)

[8.4. Software item request definition 19](#_Toc2875238)

[9. Data definition 20](#_Toc2875239)

[10. User interface definition 20](#_Toc2875240)

[10.1. Screen transition diagram 20](#_Toc2875241)

[10.2. Screen request definition 20](#_Toc2875242)

# Purpose

To describe the software architecture specifications of binocular refractometer device Becky.

# Scope of Application

This document is applicable to the software architecture specifications of binocular refractometer device.

# Definition/References

## Reference

* Becky System architecture specification　Version 1（）
* Becky Software requirement specification Version 1（GABV\_A00002）

## Definition

| Terminology and Abbreviations | Description |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

# Software Architecture Overview

## Architecture schematic diagram

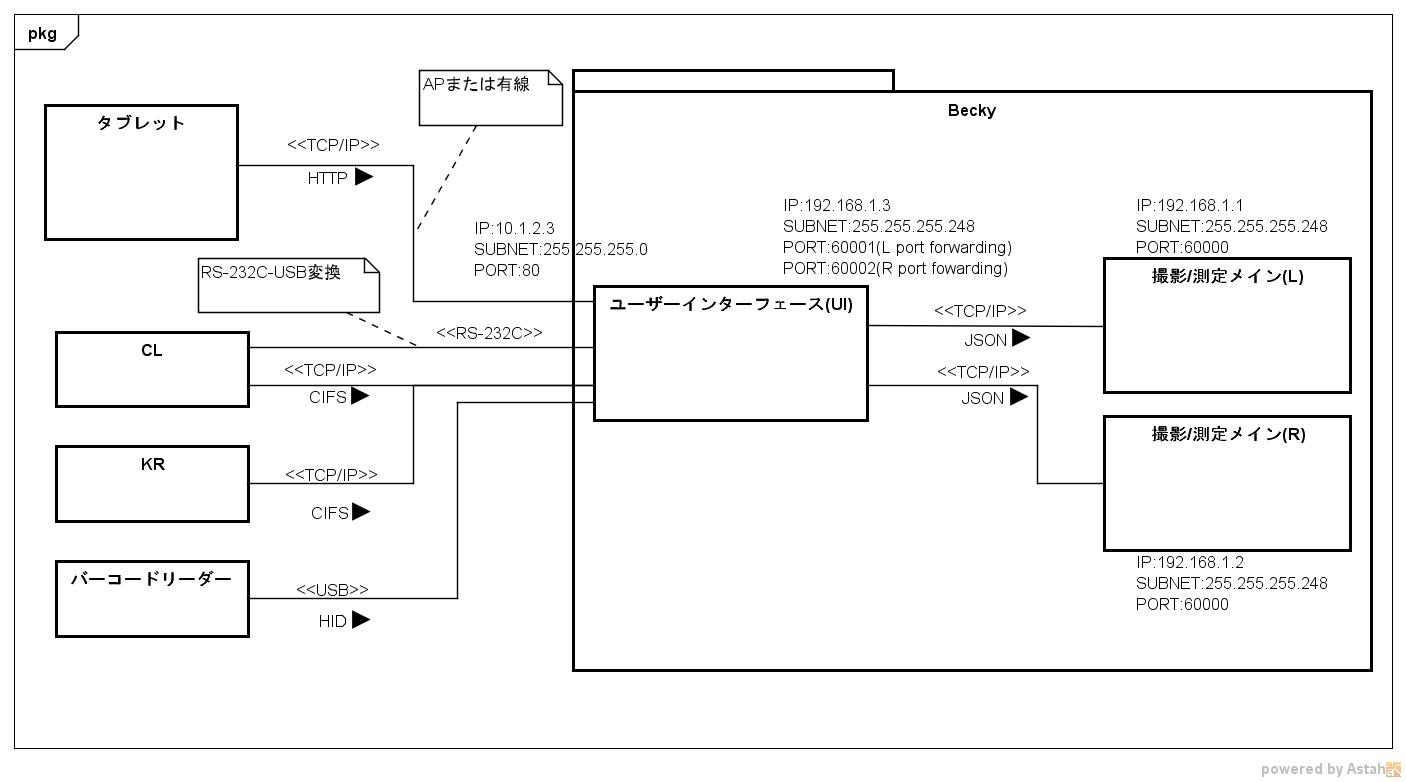
### Becky system overview

The Becky system consists of a user interface (UI) and imaging/ measurement main (R) / (L).

The user operates the GUI displayed on the Wi-Fi connected tablet to control the Becky system.

The user interface (UI) accepts an operation from the user, and sends instructions to left and right imaging/ measurement main respectively.

In the externally connected device, measurement data is input from the CL (lens meter) and KR (refract keratometer) to the shared folder (CIFS). Also, exam subject information is input from the USB barcode reader.



RS-232C-USB conversion

AP or fixed line

Barcode

reader

Imaging/Measurement main (R)

Imaging/Measurement main (L)

User interface (UI)

Tablet

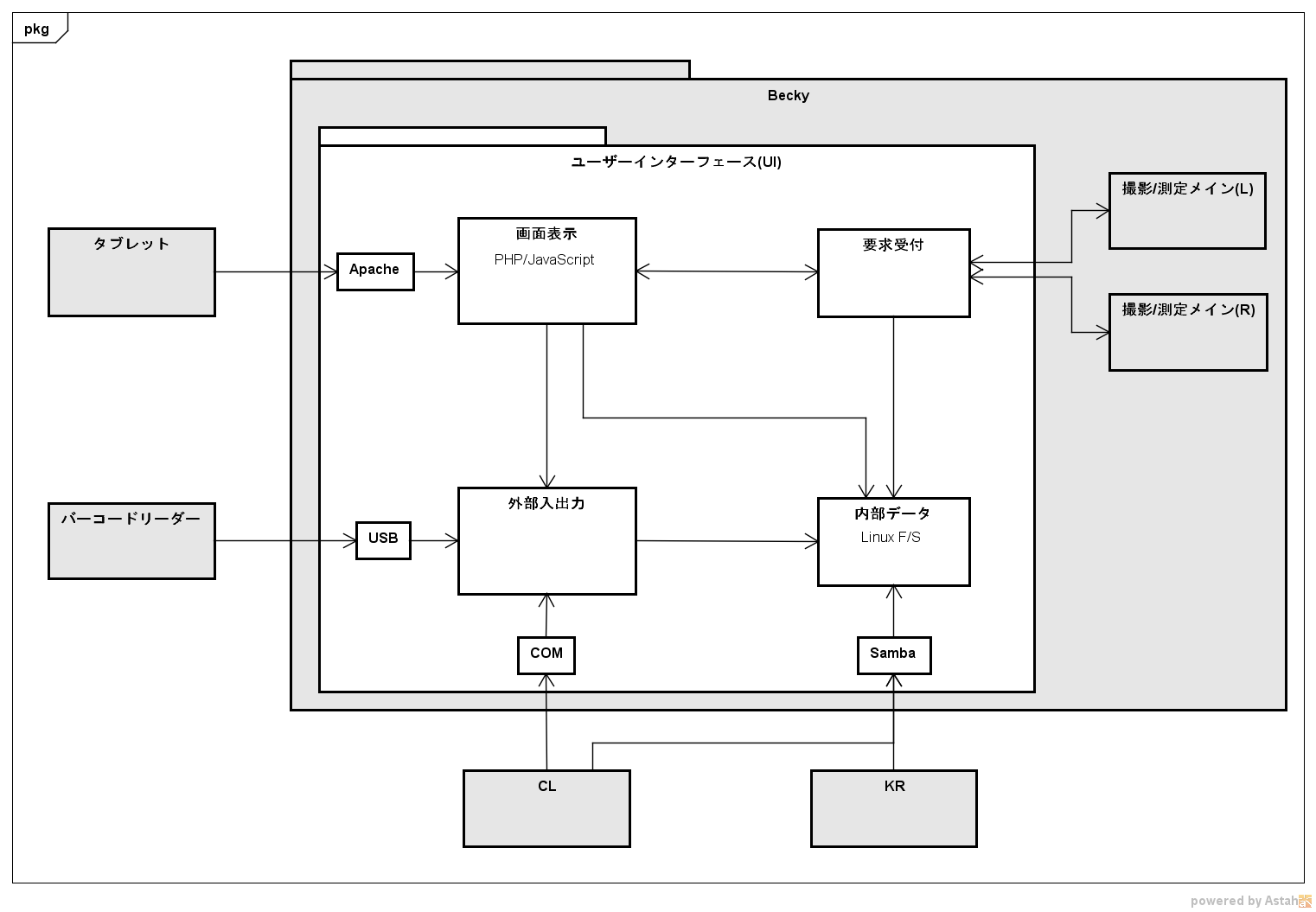
### External interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sending source | Sending destination | interface | Protocol | Description |
| Tablet | User interface(UI) | TCP/IP | HTTP | Access web server within Becky and perform operation from GUI displayed on tablet. |
| CL | User interface(UI) | TCP/IP | CIFS | Save lens meter measurement data to Becky internal shared folder |
| RS-232C | Independent | Save lens meter measurement data in original format with serial communication (RS-232C) |
| KR | User interface(UI) | TCP/IP | CIFS | Save Refkera measurement data in a shared folder inside Becky |
| Barcode reader | User interface(UI) | USB | HID | Enter exam subject information by keyboard interface (HID) |

### Internal interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sending source | Sending destination | Interface | Protocol | Description |
| User interface(UI) | Scanning measurement main(R)/(L) | TCP/IP | JSON format/Independent | Receives user operation and gives control instructions to Scanning measurement main (R) / (L) |

## User interface (UI)



User Interface (UI)

Scanning/Measurement Main (R)

Scanning/Measurement Main (L)

Bar code reader

External I/O

**Internal data**

Linux F/S

Accept request

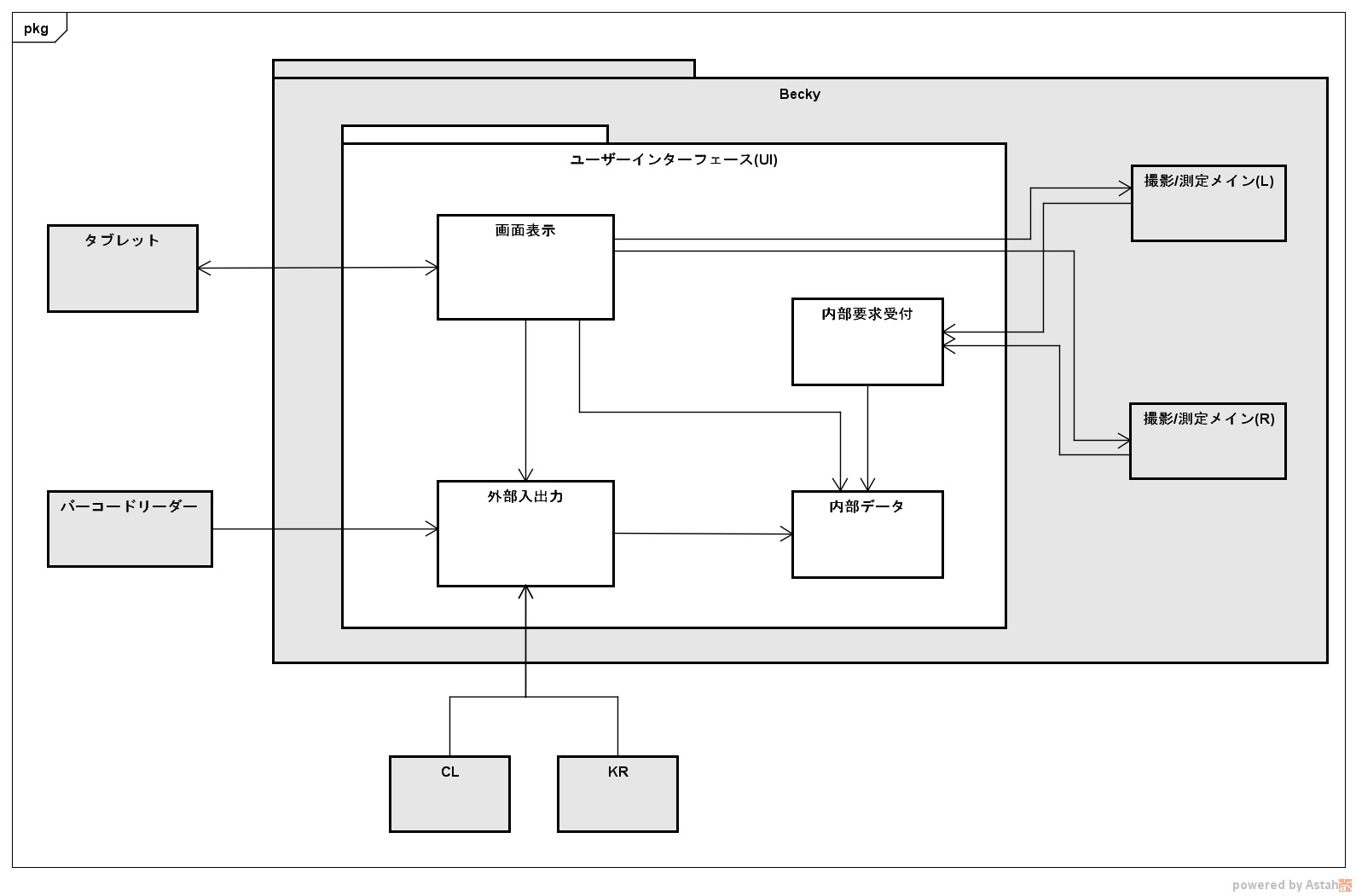
Screen display PHP/JavaScript

Tablet

### Internal interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sending source | Sending destination | Interface | Protocol | Description |
| Screen display | Receive requirement |  | JSON format/Independent |  |
| Screen display | External I/O |  | JSON format/Independent |  |
| Screen display | Internal data | F/S | JSON format |  |
| Receive requirement | Internal data | F/S | JSON format |  |
| External I/O | Internal data | F/S | JSON format |  |

## Scanning/Measurement main



Bar code reader

Scanning/Measurement Main (R)

Scanning/Measurement Main (L)

tablet

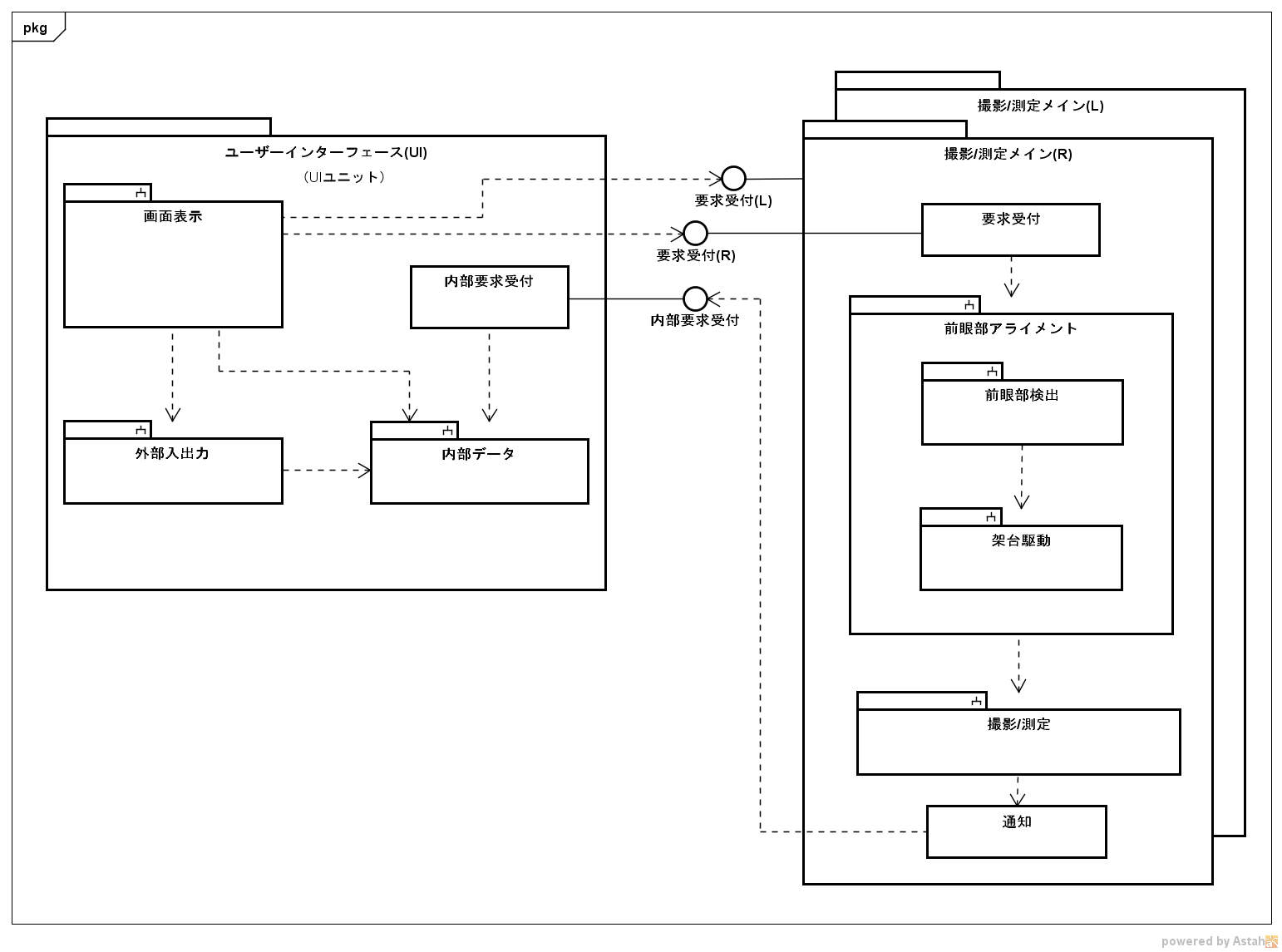
External I/O

Internal data

Receive Internal request

Screen display

User Interface (UI)



Notify

Scanning/Measurement

Gantry drive

Anterior ocular segment

detection

Anterior ocular segment alignment

Receive Internal request

Receive request (R)

Receive request (L)

Receive request

Scanning/Measurement Main (R)

Scanning/Measurement Main (L)

Internal data

External I/O

Receive Internal request

Screen display

User Interface (UI)

(UI unit)

### User interface（UI）

|  |  |
| --- | --- |
| Screen display | * Display screen for operating Becky * Notify the user operation to Scanning/Measurement main, External I / O * Display information of Internal data |
| Internal data | * Store exam subject information * Store measurement data * Store device setting values |
| External I/O | * Output measured data to external equipment with user operation * Input data from connected external input device is stored in Internal data |
| Receive Internal requirement | * Receive reply and report from Scanning / Measurement main * Save the received report in Internal data |

### Scanning/Measurement main（main CPU unit）

Scanning/Measurement main exists corresponding to left and right eye respectively.

|  |  |  |
| --- | --- | --- |
| Receive requirement | | * Accept instructions from Screen display (external receive requirement) * Accept requests from connected devices (internal receive requirement) |
| Anterior ocular segment | Anterior ocular detection | * Detect XYZ position of measurement head by stereo matching   + Pupil center   + Cornea reflection image bright spot |
| Gantry driven | * The gantry is driven from the XYZ position obtained by the anterior segment detection to a position that can be scanned * Receives the movement instruction via the user operation, drives the gantry in the designated direction |
| Scanning/Measurement | | * Measures the objective refractive power (Ref) and calculate the refractive power of the eye to be examined * Measures the corneal curvature (kerato) and calculate the corneal curvature of the eye to be examined * Performs subjective refractive power test to calculate the subjective refractive power of the eye to be examined |
| Notification | | * Responds to external requests * Reports the received internal requirement to User interface Report * Reports measurement result to the User interface Report * Reports Scanning / Measurement main to the User interface |

Becky software system consists of software items located on the following CPU board. The main CPU board (Slave) has two measurement head configurations corresponding to the scanning/measurement the left and right eye respectively.

1. Main CPU board(Master)

Control of the entire Becky system, Screen display, External I / O.

Linux (Ver. X.xx) is installed in the OS and realized as a Linux application.

Screen display is realized as a Web application running on Web Server (Apache Ver.x.xx). Main CPU board

(Slave) controls the measuring head and delivers internal status such as measurement result by Socket

communication with TCP / IP interface.

1. Main CPU board(Slave)

Accepts an instruction from the main CPU board (Master), realizes the anterior ocular segment function and the scanning /measurement function of the measuring head. Linux (Ver. X.xx) is installed in the OS and realized as a Linux application.

Performs light source and gantry drive during auto alignment and measurement by communicating with Optical sub CPU board and base sub CPU board that control the measuring head via the CAN interface.

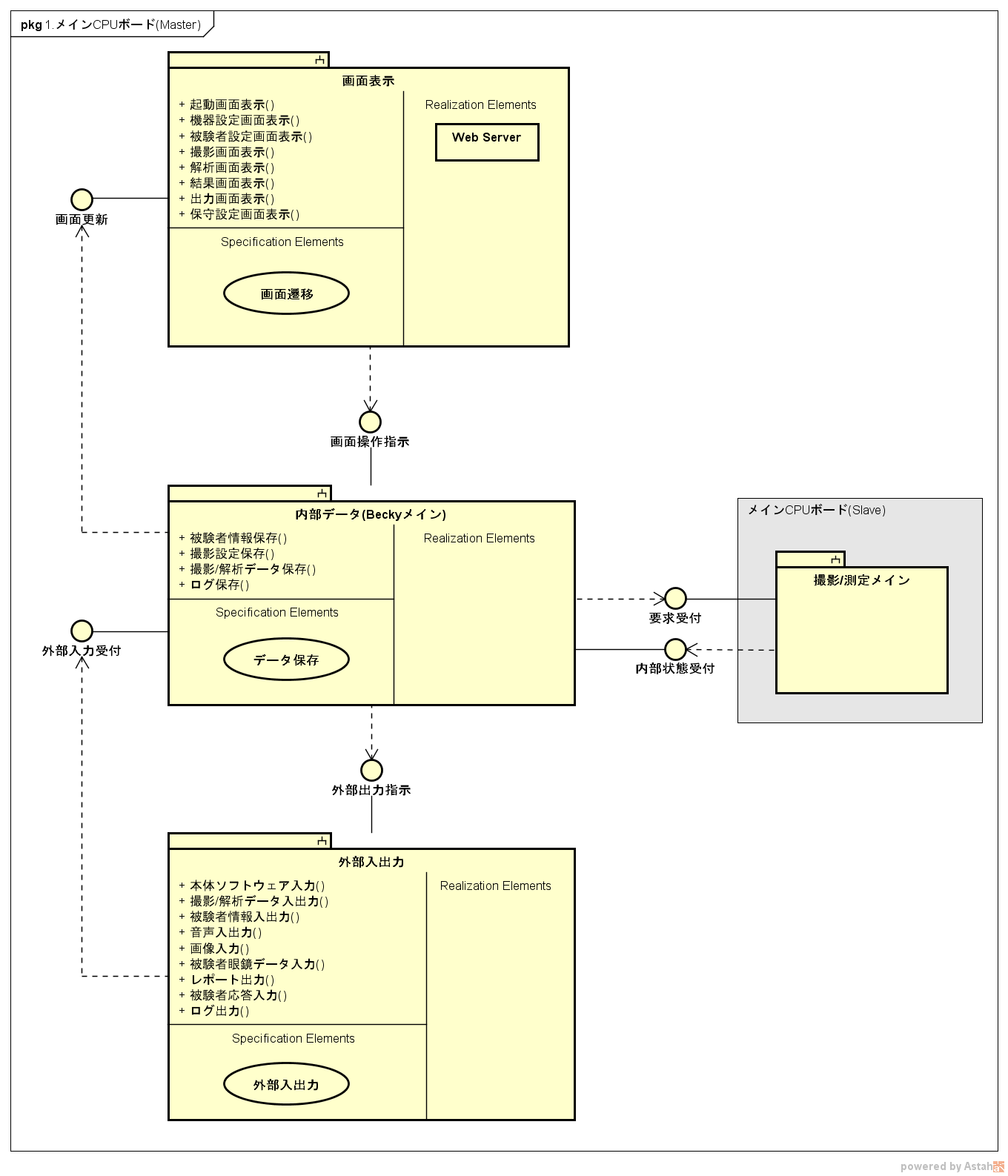
1. Optical sub CPU board

Accepts instructions from the main CPU board (Slave) and performs device control for measurement connected to the board. Mounts RTX (Ver. X.xx) on the OS and realizes it as firmware running on CMSIS.

1. Base sub CPU board

Accepts instructions from the main CPU board (Slave) and performs device control for the gantry connected to the board. Mounts RTX (Ver. X.xx) on the OS and realize it as firmware running on CMSIS.

### Main CPU board（Master）



+Input Main unit software()

+Input scanning/analysis data ()

+Input exam subject information ()

+Sound I/O ()

+Input screen ()

+Input exam subject glasses ()

+Output report ()

+Input exam subject response ()

+Input log ()

External I/O

Main CPU board (Slave)

Internal data (Becky main)

+Store exam subject information ()

+Store scanning settings ()

+Store scanning/analysis data ()

+Store log ()

Scanning/measurement main

External output instruction

Receive internal status

Receive request

Receive external input

Screen operation instruction

Update screen

Pkg1.Main CPU board (master)

+Display launch screen ()

+Display device settings screen ()

+Display exam subject setting screen()

+Display scanning screen ()

+Display analysis screen ()

+Display results screen ()

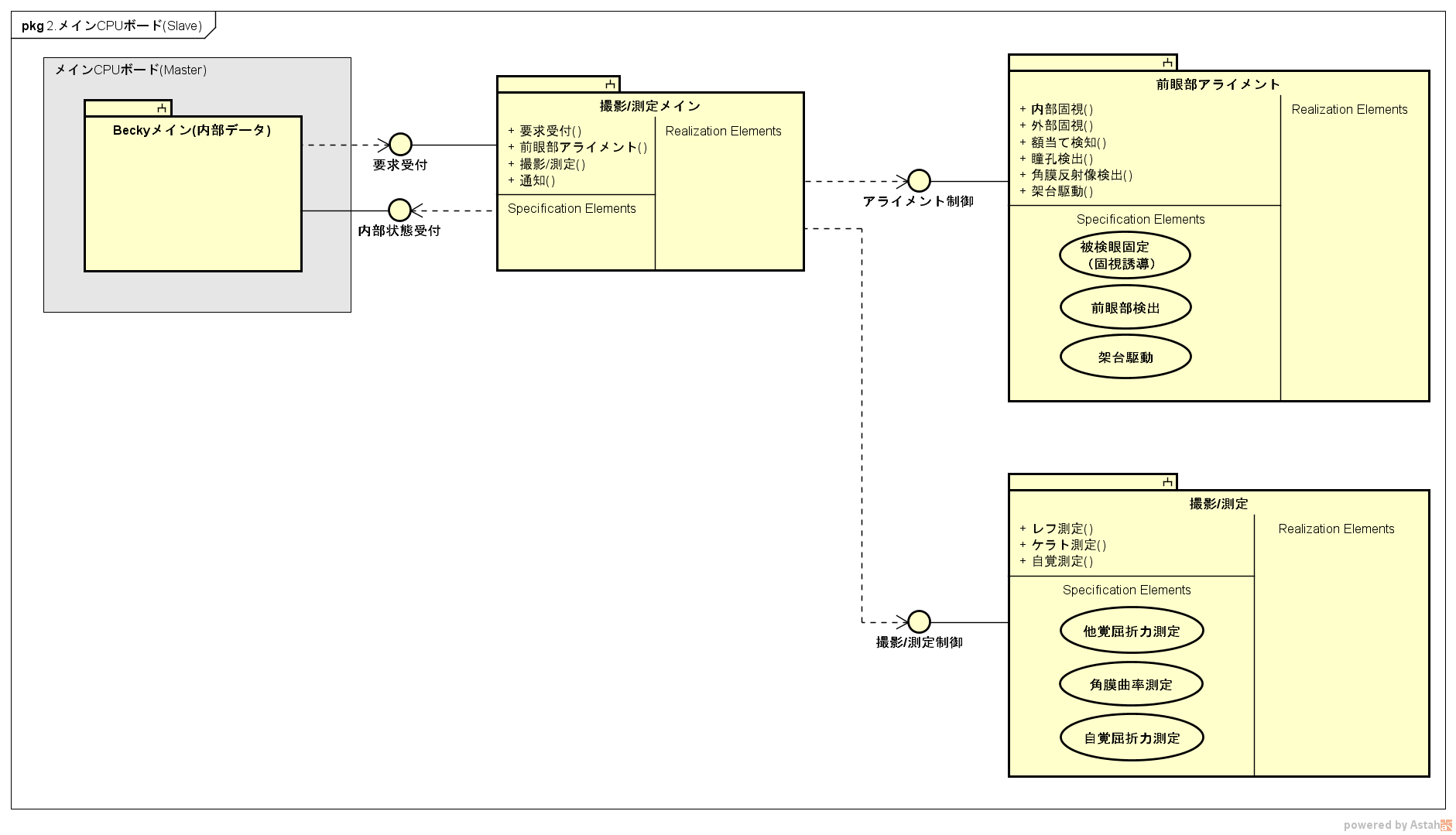
+Display output screen ()

+Display maintenance setting screen()

Screen display

|  |  |
| --- | --- |
| Software item | Functional overview |
| Screen display | * Notifies the Becky main application of control of various screen transitions for receiving input from the user and accepted inputs. * Update the display screen in response to the screen update instruction from the Becky main application |
| Internal data | * Instructs Scanning/Measurement main according to user input notified from Screen display. * Receives change notification of internal state from Scanning/ Measurement main, and updates internal data and sends updated instructions to Screen display. * Instructs the external I/O to output data in response to an external output instruction from the screen display. * Accepts data input from External I/O and sends updated instructions of internal data and update to Screen display. |
| External I/O | * In response to an instruction from the Becky main, data is output to the corresponding interface (USB, LAN) in the specified format. * Accepts input from an externally connected device and transmits input data to Becky main. |

### Main CPU board（Slave）



Receive request

Scanning/measurement main

+ Receive request ()

+Anterior ocular segment alignment

+Scanning/Measurement ()

+Notification ()

Anterior ocular segment alignment

Subject’s eye

fixation

(Visual guidance)

Anterior ocular

Segment detection

Receive internal

status

Pkg2.Main CPU board (Slave)

Main CPU board (Master)

Gantry

drive

Scanning/measurement control

+Refractrometer measurement ()

+Keratometer measurement ()

+Subjective measurement ()

Objective refractive

power measurement

Corneal curvature radius measurement

Subjective refractive

power measurement

+Internal fixation

+External fixation

+ Head detection

+Pupil detection

+Corneal reflex image detection

+ Gantry drive

Alignment control

Becky main (Internal data)

Scanning/Measurement

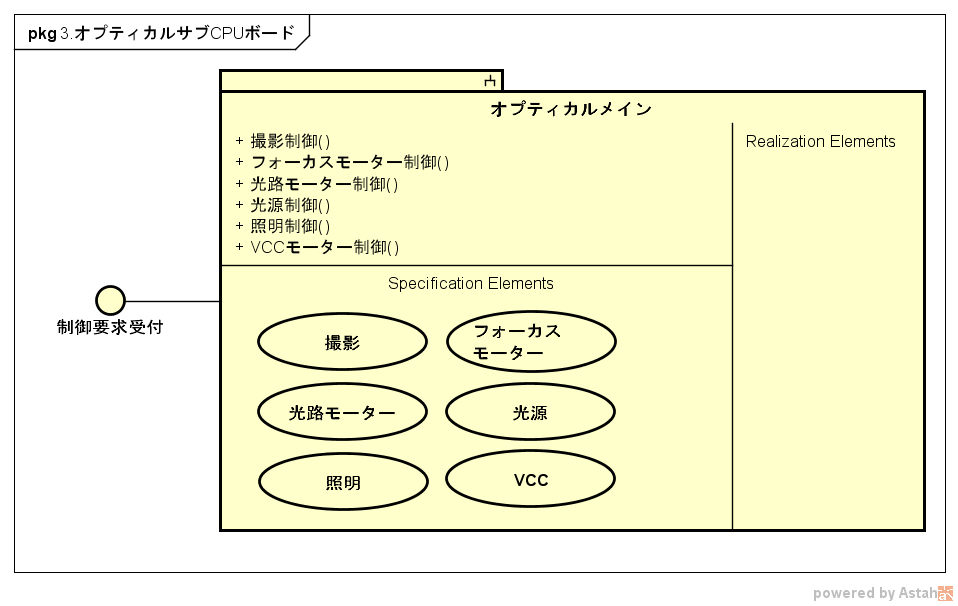
### Optical sub CPU board

Optical path

motor

|  |  |
| --- | --- |
| Software item | Functional overview |
| Measurement/Scanning main | * Anterior ocular segment alignment and scanning/ measurement are performed according to instructions from Becky main application. * Receives the result (control instruction result, state change) from anterior ocular segment alignment and scanning/measurement, and notifies Becky main application. |
| Anterior ocular segment  alignment | * "Detection of anterior ocular segment" for detecting the relative position with the measurement head is performed from the acquired anterior segment image, and auto alignment is performed by "gantry drive" |
| Scanning/Measurement | * Anterior ocular segment is scanned at completion and the analysis result is calculated from the acquired image. * Calculated analysis result is notified to "Internal data (Becky main)". |

### Optical sub CPU board



Pkg3.Optical sub CPU board

Receive control request

Illumination lamp

Scanning

Focus motor

Light source

VCC

Optical main

Optical path

motor

+ Scanning control ()

+Focus motor control ()

+Optical path motor control ()

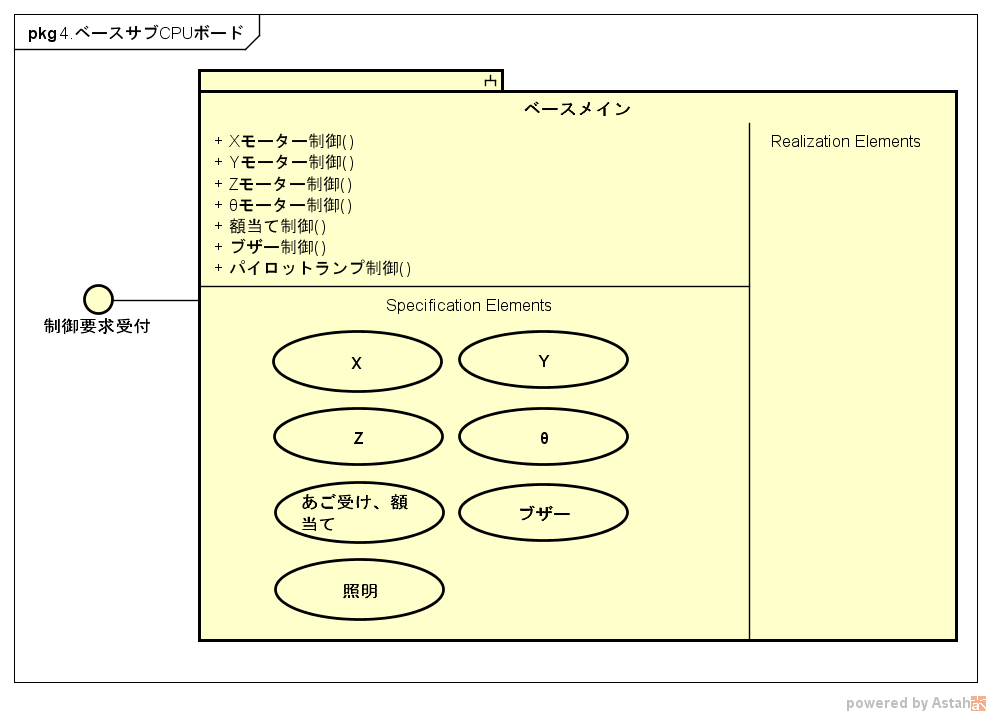
+Light source control ()

+Illumination lamp control ()

+ VCC motor control ()

|  |  |
| --- | --- |
| Software item | Functional overview |
| Optical Main | * Controls the hardware connected to the optical CPU board according to instructions from anterior ocular segment alignment and scanning/measurement. |

### Base sub CPU board



Baseline

+X motor control ()

+Y motor control ()

+ Z motor control ()

+ Ɵ motor control ()

+Headband control ()

+Buzzer control ()

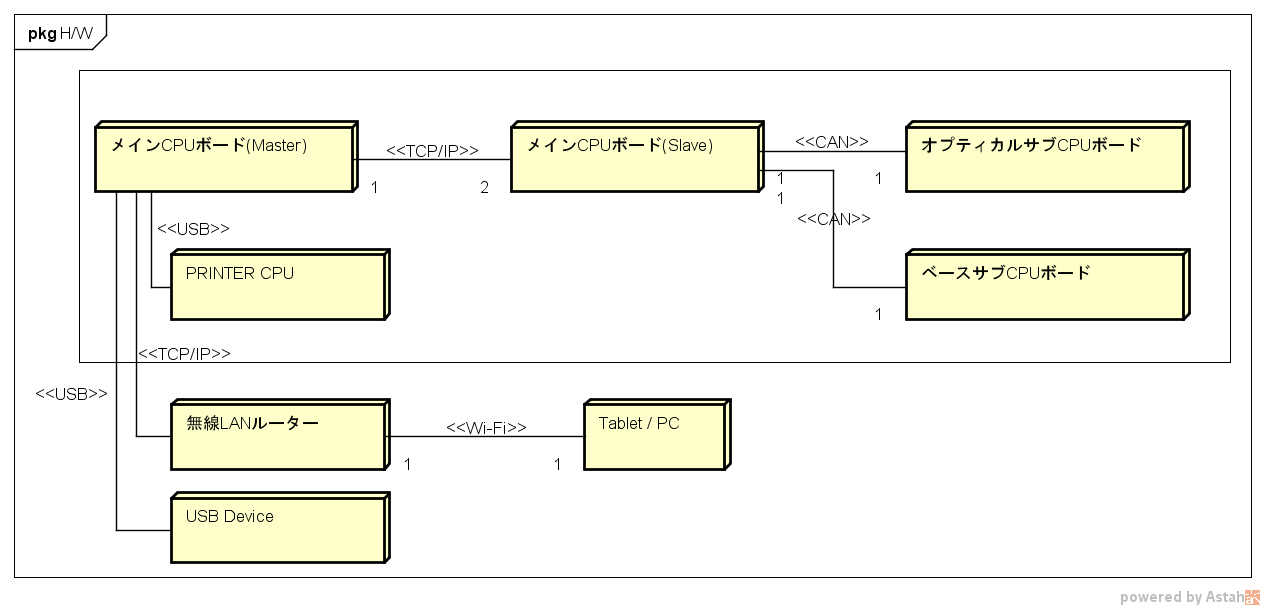
+Pilot lamp control ()

Pkg4.base sub CPU board

Receive Control request

|  |  |
| --- | --- |
| Software item | Functional overview |
| Base main | * Controls the hardware connected to the base sub CPU board according to instructions from the main CPU board (Master) and anterior eye part alignment. |

## System architecture



Base sub CPU board

Wireless LAN router

Main CPU board (master)

Main CPU board (slave)

Optical sub CPU board

|  |  |
| --- | --- |
| Interface | Content |
| Socket communication | Perform communication between Main CPU board (Master) and Main CPU board (Slave). Notifies the operation entered from Screen display |
| CAN | Perform communication between the main CPU board (Slave) and the Optical sub CPU board, base sub CPU board. Performs illumination/Lighting, light source control, and gantry drive at the time of anterior ocular segment alignment and scanning/ measurement |
| USB | 1) Output the print data to the Inbuilt printer  2) Input and output with externally connected USB devices (Barcode reader, USB memories) |
| Wi-Fi | Access the Web server running on the main CPU board (Master) via an externally connected wireless LAN router, and operate the device from the Web browser on Tablet or PC |

## Software architecture

### Main CPU board (Master/Slave)

|  |  |  |  |
| --- | --- | --- | --- |
| Web browser (touch panel) |  | Becky Main (PHP) | External I/O |
| OS※ |  | Web Server |
| Tablet |  | OS(Linux) | |
|  |  | Main CPU board(Master/Slave) | |

※Decided according to recommended tablet selection

### Sub CPU board (optical / base)

|  |
| --- |
| Sub CPU Firmware |
| OS(RTX) |
| CMSIS |
| Cortex-M4 |

# Development environment

## Main CPU board（Master/Slave）

1. Web application

|  |  |
| --- | --- |
| Development device | x86 PC |
| OS | Windows 7 Professional SP1 |
| IDE | － |
| Language | PHP, JavaScript |

1. Other

|  |  |
| --- | --- |
| 1. Development device | x86 PC |
| OS | Ubuntu 10.04 LTS（VMWare Workstation 12 Pro） |
| IDE | Eclipse x.x.x |
| Language | C/C++ |

## Sub CPU board (optical / base)

|  |  |
| --- | --- |
| Development device | x86 PC |
| OS | Windows 7 Professional SP1 |
| IDE | Texas Instruments, Code Composer Studio Version 6 |
| Language | C/C++ |

# Operating environment

## Main CPU board（Master/Slave）

|  |  |
| --- | --- |
| Target | ARM board |
| CPU | Texas Instruments, DM8148 ARM Cortex-A   * ARM Cortex-A8 (MPU) up to 1GHz * C674x Floating Point DSP Core up to 750MHz |
| memory | ARM: 32KB L1P Cache, 32KB L1D Cache, 256KB L2 Cache  DSP: 32KB L1P Cache, 32KB L1D Cache, 256KB L2 Cache  DDR3-800 2GB |
| SOUP | * Linux kernel Ver.x.xx * DVR RDK * Apache |

## Sub CPU board (optical / base)

|  |  |
| --- | --- |
| Target | Optical sub CPU board, base sub CPU board |
| CPU | Texas Instruments, LM4xxxx |
| memory | Inbuilt RAM |
| SOUP | * CMSIS * RTX |

# Design restrictions

*＜Describes design limitations. Provides details of software limitations described in software requirement specifications.*

*Also describe changes expected to occur in software and future addition of functions*

* *Software development problems and risks*
* *Unresolved points and assumptions unclear at the time of design*
* *Description of additional functions＞*

# Software item definition

## Software item list

*＜Analyze the software request described in the software requirement specification, assign the function to the software item, and create the list. Include SOUP items in the list＞*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SDS ID | Software item | Description | Safety class classification | Remarks | SRS ID |
| SDS-XXX-001 | *Software item １* |  |  |  | SRS-XXX-01 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

### ＜Software item１＞

*＜When the software item has a hierarchical structure, a list is created for each software item*

*＞*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SDS ID | Software item | Description | Safety class classification | Remarks | SRS ID |
| SDS-XXX-001-01 | *Software item* |  |  |  | SRS-XXX-01 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Software safety class classification

*＜Describes the safety classification of the software system. Describes the most risky safety class in the specified software item.＞*

Example) The software safety classification of this software system is "C".

## Software item interface definition

*＜Define interface between software items＞*

### Software item structure

*＜Indicates the static structure between software items using package (class) diagram, functional structure diagram, hierarchical diagram＞*

### Software item mutual relationship

*＜Use sequence diagrams, activity diagrams to show dynamic relationships between software items＞*

## Software item request definition

*＜Describe the input/output, processing/algorithm, non-functional requirement of each software item, state transition table/figure as and when required＞*

### ＜Software item１＞

|  |  |
| --- | --- |
| Software item name | *＜Software item１＞* |
| Overview |  |
| Input |  |
| Output |  |
| Processing / Algorithm |  |
| Non-functional requirement |  |

*＜When the software item has a hierarchical structure, a request definition is created for each software item＞*＞

# Data definition

*＜Data definition with shared use＞*

# User interface definition

## Screen transition diagram

*＜Screen transition diagram＞*

## Screen request definition

*＜Each screen specification＞*