API design

IQ Toolkit Calibration Plate Changer

Opdrachtgever	:	Philips Health Tech
Date	:	23-03-2017
Project Number	:	1

Sign client :

Version

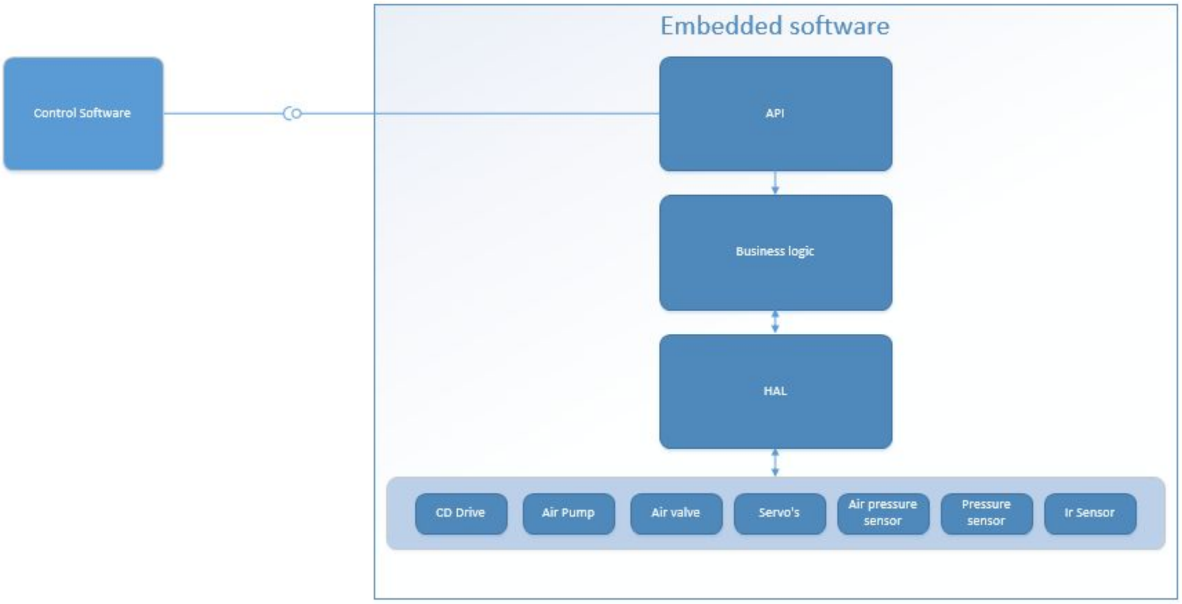
Version	Date	Author(s)	Changes	Status
1.0	23-03-2017	MT, HH	Document creation	In progress
1.1				
1.2				
1.3				
1.4				
1.5				

Distribution

Version	Date	To

API Design

The API consists an interface that offers functionality that the system’s embedded side will offer to the computer system. Between the control software and the embedded software, the communication is blocking. This means that when an operation is sent, the control software waits for a success signal before it continues sending the next one. Handling these messages is a responsibility of the user interface code.



API Overview

API Functions

These functions are offered by the embedded system, which the UI can call.

```
Result MoveTo(Position sourceIndex, Position destIndex);
std::string AuthVersion(std::string myVersion);
Result CancelCurrentOperation();
Result EmergencyStop();
Result ContinueSystem();
Result ResetSystem();
```

Functions the API offers

Custom types

In order to keep the API simple, enums will be used as custom types. In this way, the intent of the operations is clearer while the underlying data types will be simple for the transport.

```
enum class Position {
      Collimator   = -1,
      Drive0       = 0,
      Drive1       = 1,
      Drive2       = 2,
      Drive3       = 3,
      Drive4       = 4
};

enum class Result {
      Success             = 0,
      ExtendFail          = 1,
      RetractFail         = 2,
      MoveFail            = 3,
      EnableVacuumFail    = 4,
      DisableVacuumFail   = 5,
      CancelFail          = 20
};
```

Enums to define Position and Result

Function table

These functions should be implemented by both systems.

Function name	Return	Arguments	Effects
MoveTo	Result	Position sourceIndex Position destIndex	Tell the embedded system to move the calibration plate at sourceIndex to destIndex. Returns: operation result
AuthVersion	std::string	std::string myVersion	Send the control program version number to see if the software and hardware are compatible. Returns: embedded system version number.
CancelCurrentOperation	Result	none	Cancels the current operation that's running on the embedded system. An operation is defined to be one complete "MoveTo" operation. Returns: OK on success, CancelFail on fail.
EmergencyStop	Result	none	Stops all operations that are running on the hardware. This means the system is in a freeze state. Returns: OK on success, CancelFail on fail.
ContinueSystem	Result	none	Removes the current error status to allow the system to continue the current operation. OK on success, ExtendFail, RetractFail, MoveFail when the operation can not continue.
ResetSystem	Result	none	Resets the entire system state and moves everything into default position. OK on success. ExtendFail, RetractFail, MoveFail when an operation can not continue.

Table of functions with their arguments, return value and a short explanation