

Business logic

IQ Toolkit Calibration Plate Changer

PHILIPS



Fontys

Hogeschool ICT

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

Sign client :

Version

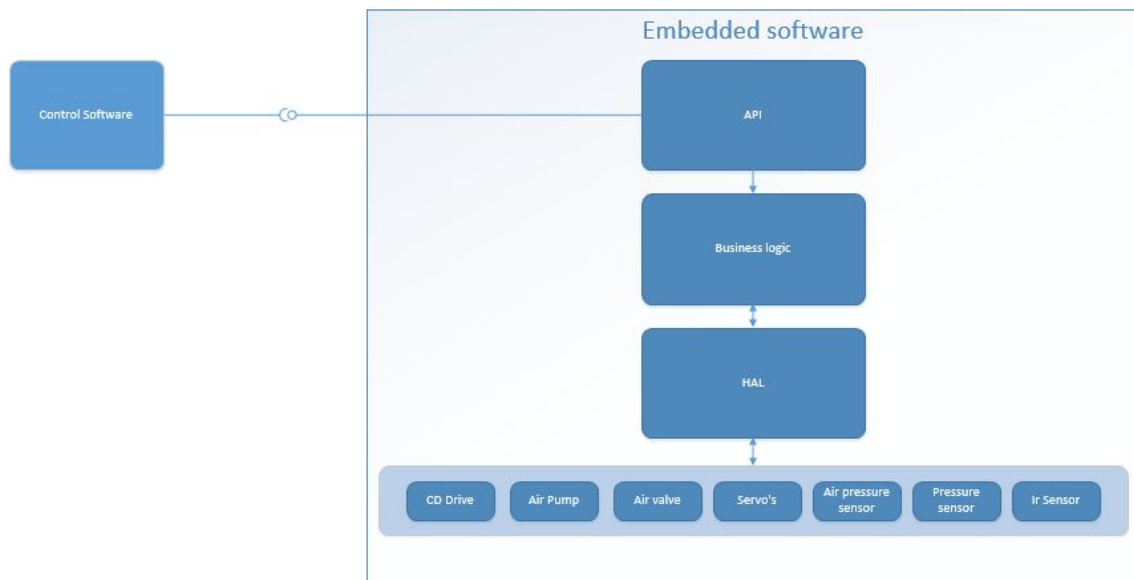
Version	Date	Author(s)	Changes	Status
1.0	24-03-2017	MT, HH,KB, RWT	Document creation	In progress
1.1				
1.2				
1.3				
1.4				
1.5				

Distribution

Version	Date	To

Business logic

The Business Logic translates the API calls to control signals for the HAL and other verification functions for the embedded system unit itself. This document describes the communication between API-level and the Business Logic, and between the Business Logic and Hardware Abstraction Layer.



There's communication between the API and the Business Logic. In the table below the communication is defined.

API Function	From Business layer
Result MoveTo(Position sourceIndex, Position destIndex);	Returns OK on a successful complete operation and a FAIL state depending on what failed from the HAL.
std::string AuthVersion(std::string myVersion);	Returns current embedded software version
Result CancelCurrentOperation();	Returns OK on successful cancellation, FAIL state on failure.
Result EmergencyStop();	Returns OK on successful stop, FAIL state on failure.
Result ContinueSystem();	Returns OK on successful continue, FAIL state on failure.
Result ResetSystem();	Returns OK on successful reset, FAIL state on failure.

There's communication between the Business Logic and the HAL. In the table below the communication is defined.

Function	From Business logic to HAL
Result OpenDrive(int drive)	Returns OK on successful open the drive.
Result CloseDrive(int drive)	Returns OK on successful close the drive.
Result EnableVacuum()	Returns OK when there is vacuum.
Result DisableVacuum()	Returns OK when there is no vacuum
Result MoveArm(Coord Coord)	Returns OK when all servo's are on the right position.

When the Business Logic layer gets the command MoveTo(Position sourceIndex, Position, destIndex) from the API layer, the program will run through the following steps to ensure that the plates are in the desired position.

Step	Function	Command
1	Move the plates from X to Y	MoveTo(Position sourceIndex, Position DestIndex)
2	Find the coordinates	Coord FindDriveCoord(Position SourceIndex); Coord FindDriveCoord(Position DestIndex);
3	Open the correct plate's drive	OpenDrive(Position SourceIndex/Position DestIndex);
4	Move the arm to the correct position	MoveArm(coord SourceCoord);
5	Turn the vacuum on	EnableVacuum();
6	Move the arm to the correct position	MoveArm(coord DestCoord);
7	Turn off the vacuum	DisableVacuum();
8	Return the robot arm to its idle position	MoveArm(coord HomeCoord);
9	Close the correct plate's drive	CloseDrive(Position sourceIndex/Position DestIndex);