1．背景(Background)

The Argonaut system was sent to pre-testing to see if the device can pass EMC testing and gather useful information to further improve the device for successfully passing the EMC tests. As the Argonaut system has several modalities combined into a single device, each modality must be tested separately.

2．目的(Objective)

To have a testing software for the Biometry mode for any impacts that can affect the output of the device while going through EMS tests. EMS tests send frequencies to the device. This will be performed to see how the Argonaut would be affected by possible surrounding frequencies that may be emitted.

3．概要(Overview)

The main focus of the EMS mode is to check if there are any impacts directly affecting the Biometry mode of the Argonaut system. This includes the electronics driving the motors, cameras, laser and the controlling software. A test sequence needs to be created to check the functionality of the components. However, during EMS, the XYZ stage is set to a specific position and will only loop through the main functionality of acquisition to check the output of the data.

4．本文(Procedure)

4.1 Place and Environment

  The implementation of the EMC Testing procedure was carried out in the SPL main lab room on the first floor of the E building at santec HQ. The final software and testing was done at JQA during the EMC testing.

4.2 Subject of the evaluation

4.2.1 Product to be evaluated

The test subject to be used for the software was Argonaut Test Model 1-3 (2408S002)

4.2.2 Equipment and Software

Computer: HP Omen Transcend 16 with the following specifications:

- Intel Core I9 CPU

- 32 GB Memory

- Nvidia RTX 4070 GPU

  Software: OctEngine.TestGui commit [adc4365]

4.3 Experimental Procedure

4.3.1 Person who conducted the experiment:

Implementation of the automation is carried out by Tony Xu. The person in charge of running the automation at JQA is Tony Xu.

4.3.2 Procedure

A third party API was used in conjunction with the OctEngine.TestGui software for Argonaut. The third party API was Robot Framework, which mimics human interactions such as keyboard presses and mouse movements and clicks. An external script was created to automate a repeated sequence that is listed below.

Test Sequence:

1. Start OctEngine.TestGui

2. Load Biometry configuration

3. Connect to microcontrollers (Arduino Nano + Ruggeduino Mega)

4. Power and connect to HSL-1 and set to 10 kHz frequency

5. Home XYZ Stage

6. Set velocities and target positions and move XYZ Stage to position

7. Home ODL

8. Set ODL speed and position

9. Move ODL to position

9. Toggle optical probe to HIGH, enable MEMS, LEDs (Anterior IR Ring LED, Fixation LED)

10. Start emitting laser

11. Start acquisition preview of Biometry

12. Start capturing data

13. Save data to disk

14. Start Corneal Reference Image capture

15. Turn off LEDs

16. Home ODL

17. Repeat steps 9 to 16 indefinitely.
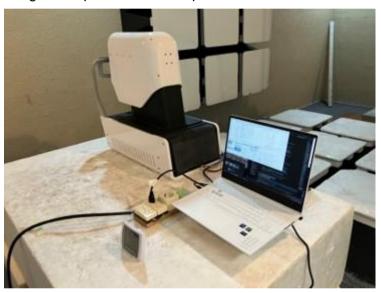

Image 1: Separate batch script to run automation

Image 2: Script running at JQA

4.3.3 Evaluation Criteria

The evaluation of the implementation is defined as to have the Argonaut Device continuously run without issues when there are no errors or issues on the Argonaut Device.

4.4 Measurement Results

4.4.1 Data collection and judgement

When there are no errors or issues with the components, the device runs the test loop until the amount of iterations are met. Generally, the amount of iterations should be more than the length of the

test. As issues arise from being exposed to different frequencies, the software errors out to indicate there was an issue with the device, whether the cameras, microcontroller or digitizer board were affected.

4.4.2 Graphs/Figures and Tables

Already included in the text.

4.4.3 Original Data

Logs and error data are located at \\File05\01_octbu\10. Medical\03. ARGOS 3\10. 実験データ (Experiment Data)\124. JQA

Software files are located in the argos3-octengine repository

5．結論・考察(Results & Discussion)

The test sequence and script allowed the device to automatically toggle components related to Biometry and capture data to see if the image data is compromised by the frequencies. This also allows the device to check if there were any electrical issues that will interfere with the software running. If an error occurs in components such as the cameras, this will stop the script and give us an error. The data recorded from the EMS testing allows us to identify the issues and improve for the official EMC testing.

6．今後の課題(Further questions & topics)

Decrease the sensitivity of detecting errors (e.g. Increase time-out error) to make the test software more robust.