DIGITAL CORE TECHNOLOGIES

Rev A0-00
10^{th} July 2024

DS_CSLI_MSEOVS
Development and Test Plan

Revision History:

Date	Version #	Description	By
10th July 2024	A0-00	Draft	Richu Bini

Contents

List of Figures

1 Development and Test Plan

1.1 Software

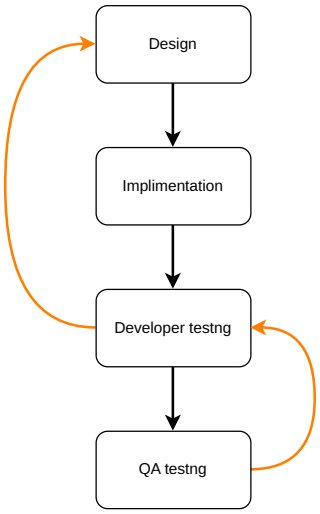

Figure 1.1. Work flow of development Phase

1.1.1 Design

This initial phase focuses on designing both the software and hardware modules for the product. Additionally, comprehensive test plans are created for each individual module to ensure thorough testing throughout the development process.

1.1.2 Implementation

During this phase, the development team translates the designs into functional software modules and hardware elements.

1.1.3 Developer Testing

Following development, each module undergoes rigorous testing by the developers themselves. Any issues identified trigger a redesign, implementation, and retesting cycle for the affected module. This iterative process ensures that each module functions as intended before being integrated with the entire system.

1.1.4 QA Testing

Once individual module testing is complete, the fully integrated system, encompassing both software and hardware, is then delivered to a dedicated Quality Assurance (QA) team for comprehensive

testing. The QA team utilizes the pre-defined test plans created during the design phase and the new test cases that the tester added to meticulously test all functionalities and features of the product. If any problems are discovered, the QA team reports them to the development team. The development team then addresses the identified issues, fixes them, and resubmits the product to QA for further testing. This iterative process continues until the product is free of defects and meets all quality standards. Upon successful completion of QA testing, the development team initiates the product release procedure.

1.2 Hardware

The following is the list of processes that we follow for the development of the hardware system in order.

1. Requirement analysis
2. HW design documentation
3. HDD review and update
4. Schematic Design
5. Schematic review and update
6. PCB Floor plan
7. PCB placement review and verification from mechanical side
8. Layout routing
9. Layout routing review and update
10. Draft Gerber release
11. Draft gerber review and update
12. Gerber release
13. PCB fabrication and assembly
14. Board bring up and testing

1.3 Detailed Development Plans With Schedules

1.3.1 Hardware

Task	Duration
Hardware Design	PDR+ 3 weeks
Schematic Design	PDR + 4 weeks
PCB Layout design	PDR + 7 weeks
Gerber Release	PDR + 8 weeks
FAB and Assembly	PDR + 14 weeks
Basic Bring up and testing	PDR + 16 weeks

1.3.2 Software

The following are the milestones and the schedules of software development and testing.
Milestone 2

Task	Duration
Detailed Software Design	PDR + 2 weeks
Preparation for trials	PDR + 4 weeks
CLASS review	PDR + 4 weeks
Setup basic camera ready for capturing the dataset	PDR + 5 weeks
PTZ and LRF interfacing	PDR + 6 weeks
Setup communication node for interaction with HMI/Custom PC application	PDR + 7 weeks
Actual trials for capturing images in different environment	PDR + 10 weeks

Milestone 3

Task	Duration
Object detection and tracking algorithms implementation phase-1	PDR + 13 weeks
Detection trial -1	PDR + 14 weeks
Object detection and tracking algorithms implementation phase-2	PDR + 16 weeks
Detection trial -2	PDR + 17 weeks

Milestone 4

Task	Duration
QA Testing in DCT	PDR + 18 weeks
QA Testing in field	PDR + 19 weeks
QA Testing at CSAS	PDR + 20 weeks

1.4 Types of Test

1.4.1 Hardware Tests

- **Baisc HW bring up and test:** Will check all the basic functionality and interface testing.
- **HW protection test:** We will test the voltage protection conditions.
- **EMI /EMC test:** Applicable for the Surface ship as per the MIL-STD-461E

1.4.2 Mechanical Tests

- **Environmental stress test :** Applicable as per MIL-810G
- **IP65 testing :** IP65 testing

1.4.3 Software Tests

- **Unit test:** Will verify that all modules are individually working by providing specific inputs and handling errors correctly.
- **Integration testing:** Will check the interactions between the SW modules and the interaction between the SU and PU to ensure they are working correctly, and verify that the software functions properly.
- **System test:** Verifies all the features are working correctly

1.5 Purpose and Expected Results of Test

1.5.1 Hardware and mechanical testing expected results

- **Baisc HW bring up and test:**
  Purpose: The bringup test is to test hardware is working correctly. It should test functionality and interface are workig correctly.
  Expected result: basic functionality and interface are workig correctly.
- **HW protection test:**
  Purpose: This test is to check the hardware is able to withstand the voltage protection conditions.
  Expected result: The hardware withstand the voltage protection conditions without any damage.
- **EMI /EMC test:**
  Purpose: This test is to ensure that electronic devices operate correctly in their intended environment without causing or being affected by unwanted electromagnetic interference.
  Expected result: That the device meets all emission limits and immunity criteria specified by MIL-STD-461E.

1.5.2 Mechanical testing expected results

- **Environmental stress test :**
  Purpose: Performed on electronic devices and systems to ensure they can operate correctly and reliably under extreme environmental conditions. These conditions can include variations in temperature, humidity, vibration, shock, and other environmental factors that the device might encounter during its lifecycle.
  Expected result: That the device meets all the above tests as per MIL-810G

- **IP65 testing :**
  Purpose: This test is to assess the degree of protection provided by enclosures of electronic and electrical devices against the intrusion of solid objects and water.
  Expected result: That the device meets all criteria specified by IP65

1.5.3 Software testing expected results

Unit testing :
The expected result of unit testing is to verify that individual software modules function correctly in isolation.

- The module produces the correct output for a given set of valid inputs.
- The module handles invalid inputs as expected (e.g., throws errors, returns specific values).
- The module behaves according to its intended design and doesn't exhibit unexpected behavior.

Integration testing :
The expected outcome of integration testing is to confirm that individual software modules function together seamlessly once combined.

- All features outlined in the project requirements work correctly across various scenarios.
- The application operates efficiently, meeting performance benchmarks for speed and stability.

System testing
The expected result of system testing is to verify the entire software application functions as a cohesive unit and meets all requirements.

- The module correctly share and interpret data.
- The module trigger each other in the correct sequence.
- Errors propagate and are managed properly to prevent crashes.
- Module interfaces communicate smoothly without conflicts.

1.6 Test Plans and Methodologies

1.6.1 Hardware

- Baisc HW bring up and test : Will check all the basic functionality and interface testing. Methodologie for testing is Manual testing
- HW protection test : We will test the voltage protection conditions. Methodologie for testing is Manual testing
- EMI /EMC test : Applicable for the Surface ship as per the MIL-STD-461E Methodology for testing is in accordance with the EMI/EMC test environment.

1.6.2 Mechanical

- Environmental stress test : Applicable as per MIL-810G Methodology for testing is as per environment stress test conditions
- IP65 testing : IP and Ingress Testing. Methodology for testing is as per IP test environment conditions

1.6.3 Software

Software development, testing includes unit testing, integration testing, and system testing. Unit testing verifies individual software modules by providing specific inputs and checking if the outputs match the expected results. Integration and system testing focus on how these modules interact and ensure the entire software behaves correctly

Unit testing :

During unit testing, each software block undergoes rigorous testing to ensure its functionality. The following test plans outline the procedures for each software module.

1. ***Target Tracker***
    - *Object Tracking:* The unit test provides an object ID as input and verifies if the camera successfully tracks the designated object.
    - *IP Joystick PTZ Control:* The test simulates a joystick command and confirms the PTZ unit (Pan-Tilt-Zoom) moves accordingly.
    - *Slew to Cue API:* The unit test injects API calls to move the PTZ to a specific location and validates its successful movement based on the provided commands.
2. ***Video Processor***
    - *Fusing video:* During unit testing, decoded daylight and thermal video are input along with the fusion percentage to verify if the output is the fused video. The process also checks how the fusion percentage changes by varying the fusion percentage values.

- *Saving video and snapshots :* This unit test verifies saving video and capturing snapshots. It checks if the video is saved in the correct format and location, and throws errors for invalid data.
- *Thermal object detection:* This unit test assesses thermal object detection. It feeds the system thermal image data and checks the output for accuracy. The test verifies if bounding boxes are correctly drawn around detected objects and classification labels are assigned correctly. It also includes scenarios with no objects, varying object sizes, and corrupted data to ensure the system functions as expected under different conditions.
- *Daylight object detecting:* This unit test assesses daylight object detection. It feeds the system daylight image data and checks the output for accuracy. The test verifies if bounding boxes are correctly drawn around detected objects and classification labels are assigned correctly. It also includes scenarios with no objects, varying object sizes, and corrupted data to ensure the system functions as expected under different conditions.
- *Metadata saving and streamin with time stamp:* The unit test ensures that the system correctly streamed metadata with timestamps, and saves metadata accurately in the desitrd location.
- *Video encoding and streaming:* This unit test verifies video encoding and streaming functionality. It feeds the system raw video data and ensuring correct encoding with designated settings, successful stream initiation using the RTSP protocol, and proper authentication before making the RTSP stream available for viewing.

3. ***Comms and Alerter***

- *Metadata Creator:* This unit test verifies that all communication with the NMEA devices is working perfectly and checks if the system can receive the object detection metadata. Then verifies that the detected object class, bounding box coordinates, confidence rate, and the position of the object relative to the radar data are added correctly.
- *Collision alerter:* This unit test will verify that the radar data is being received correctly. It will then simulate a collision detection scenario and check whether the collision is detected accurately.

Integration testing :
Focuses on how these modules work together, ensuring they interact smoothly once combined. Integration testing examines the interaction between software modules and also the SU to PU intractions, ensuring they work together smoothly.

- *Data Exchange:* Verifying modules share data correctly and interpret it properly.
- *Control Flow:* Ensuring modules trigger each other's functions in the correct sequence.
- *Error Handling:* Checking if errors are properly propagated and managed to prevent cascading failures.
- *Interface Compatibility:* Confirming that modules communicate through defined interfaces without conflicts.

System testing
Examines the complete application as a whole, verifying it functions as intended and meets all requirements.System testing acts as the final quality check for the entire software application.system testing identifies and addresses any critical issues before the application is deployed to real users. In this test all the fearures of the product are tested and verified.

Methodologies
All the software testing are done manually.

1.7 Test Reporting

The test reporting will be comprehsnsive document that contain all the test cases and the test report that will be added afther the QA testing.

1.8 Testing and Compliance

To be done along with mechanical info updation