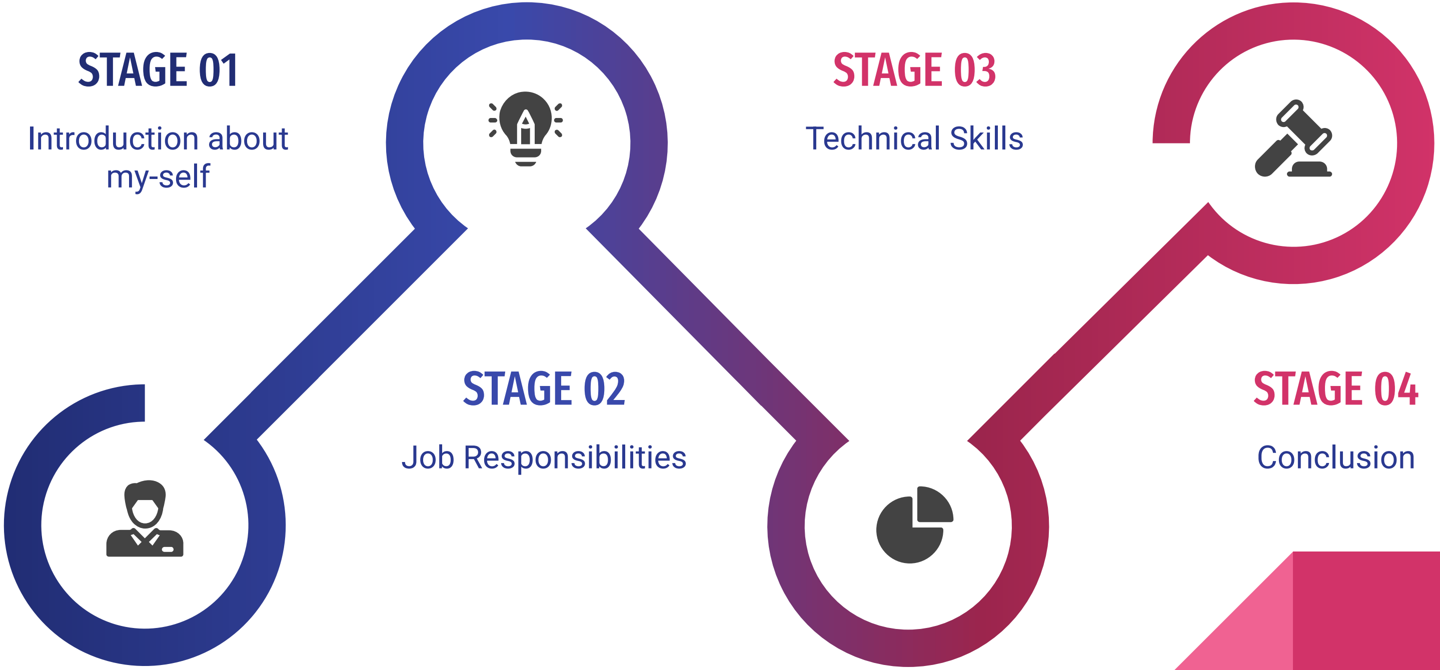
Introduction

My Roles and Responsibility

Agenda



STAGE 01

Introduction about my-self

STAGE 02

Job Responsibilities

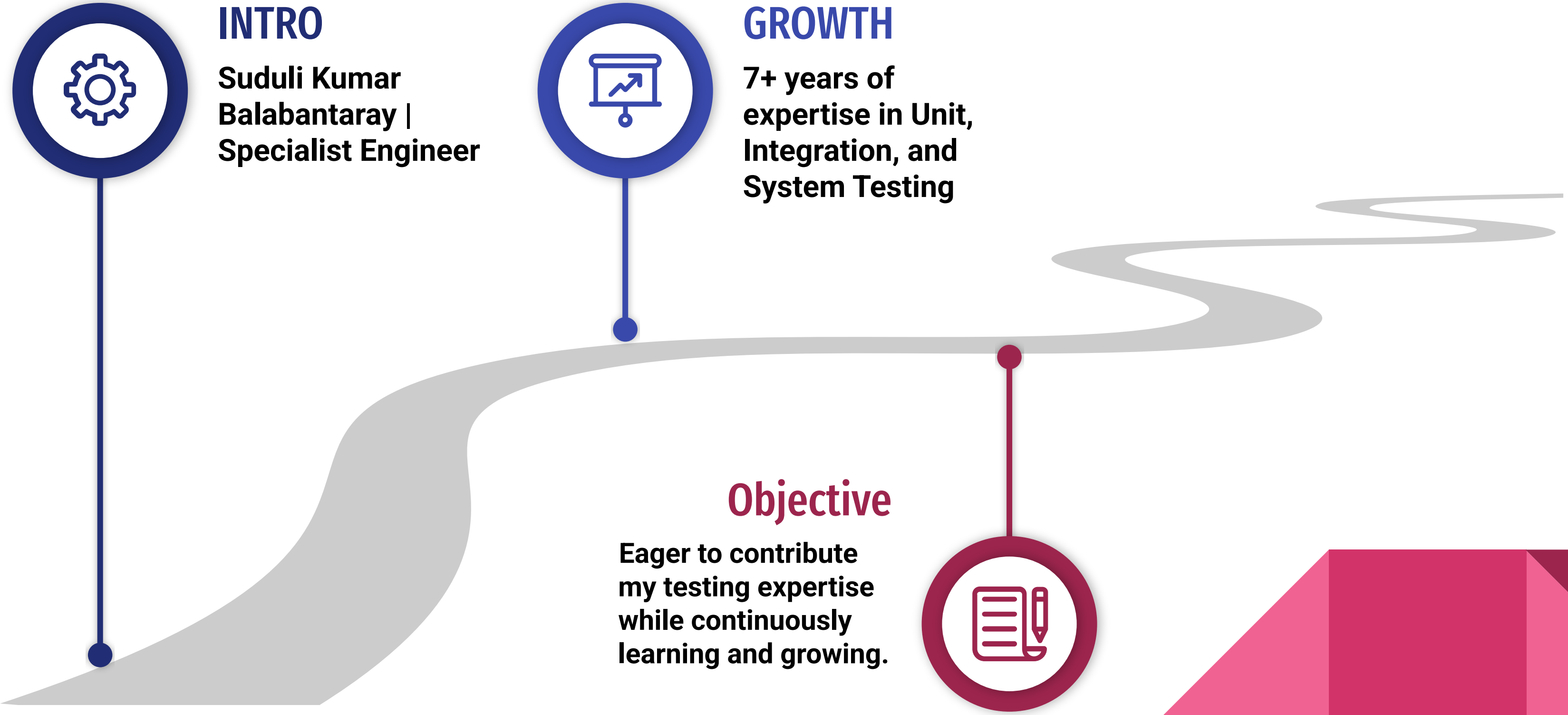
STAGE 03

Technical Skills

STAGE 04

Conclusion

Basic Details



INTRO

Suduli Kumar Balabantaray | Specialist Engineer

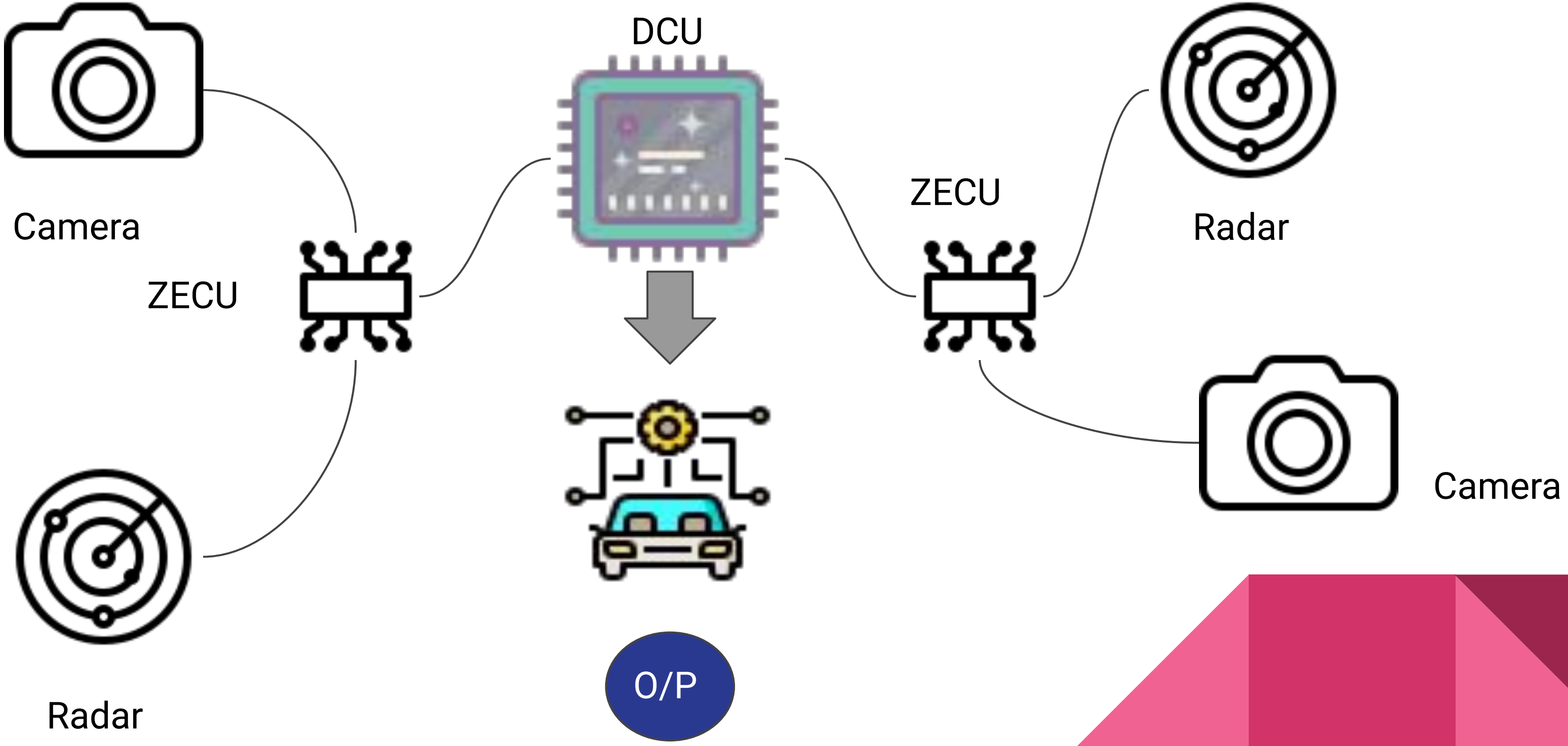
GROWTH

7+ years of expertise in Unit, Integration, and System Testing

Objective

Eager to contribute my testing expertise while continuously learning and growing.

DCU Functionalities



Responsibilities:Input Files

1.Engage with clients to obtain essential input files and initiate project execution

2.Upon receipt, validate the input files provided by the client.

- Master Requirement Sheet
- DBC Files
- Executable File
- RQMT File
- CDD and DLL file

Responsibilities:Hardware Setup Checklist

JTAG Connector

Ensure it aligns with the specific hardware requirements

Trace32 Condition

Verify its proper functioning and readiness.

Crocodile and Harness Cable

Inspect for any damages or loose connections.

CAN Case XL

Confirm the correct model (1640, 1630, or 5610) is being used

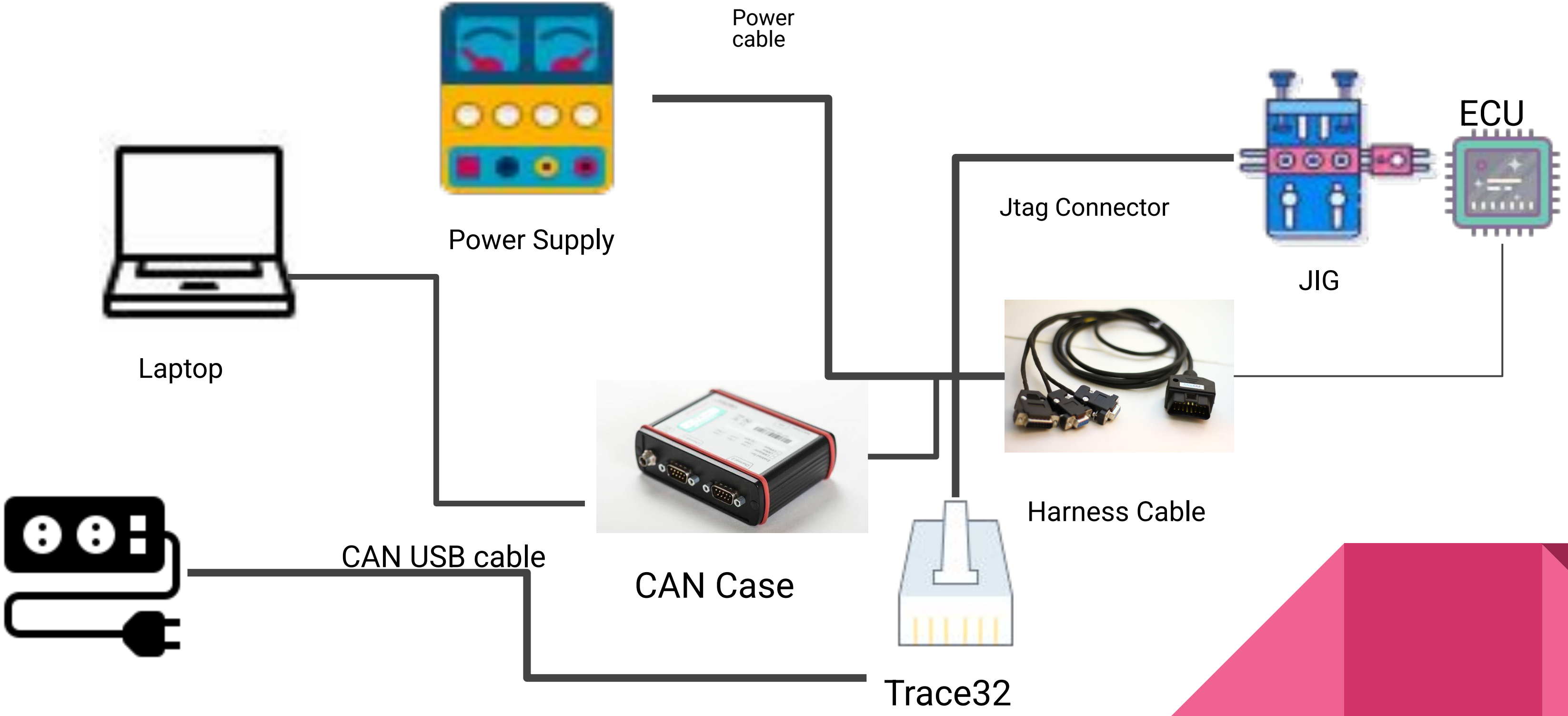
Variable Power Supply

Check its condition and ensure it's set to the correct voltage/current

JIG Pin Connections

Ensure DB9 and 120 ohm connector they are correctly set for flashing.

Responsibilities:Bench Setup



Responsibilities:Environment Creation

1.Create a simulation for Camera and Radar inputs.

01
02
03
04
05
06

DBC Integration

Integrate all relevant DBC (Database CAN) files, ensuring they contain the necessary message and signal definitions

Environment Variables

Add the required environment variables within CANoe for proper simulation functioning.

CAPL Code Creation

Write the necessary CAPL code to simulate camera and radar inputs.

CANoe Panel Creation

Design a user-friendly panel in the CANoe tool for easy simulation control and monitoring

Linking Environment Variables

Connect the environment variables to the CANoe panel to ensure real-time data flow and control.

Simulation Execution

Run the simulation in CANoe, monitoring the Camera and Radar inputs for accuracy and desired behavior.

Responsibilities:Smoke Testing

Initialization

Ensure the DCU system powers up correctly without any errors. Verify that all sensors (cameras, radars, LiDAR) are initialized and detected

DTC

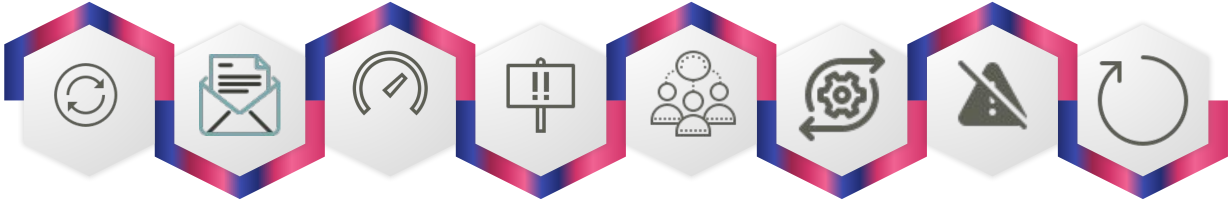
Use the 'Read DTC' service to fetch DTC status and Ensure that DTCs can be read without system crashes or delays.

Messages

Confirm that the DCU processes and responds to these messages correctly.

Zero DTC

Lorem ipsum dolor sit amet.. Donec gravida mi quis odio auctor, sed accumsan JustFreeSlide.com. Donec gravida mi quis odio auctor.



Communication

Confirm that the DCU is communicating effectively with all connected sensors. Check for any communication errors or delays in the CANoe tool.

Requirement

Quickly validate a few critical requirements from the Master Requirement sheet to ensure they're being met.

Recovery

Introduce a basic error or fault and ensure that the DCU can identify and attempt to recover from it.

Restart

On restart, confirm that no errors are introduced and the system returns to its last known good state.

Responsibilities:Achieving Zero DTC

- **CANoe Simulation:** Initiate the CANoe simulation process.
- **UDS Extended Mode:** Transition to the Extended Mode within UDS.
- **DTC Service Check:** Utilize the 'Read DTC' service to determine the DTC status and count.
- **Master Requirement Cross-Check:** Validate the identified DTCs against the Master Requirement sheet.
- **Message & Signal Identification:** Determine which messages and signals are triggering the DTCs.
- **DTC Resolution:** Implement recovery conditions to resolve the identified DTCs.
- **Achieving Zero DTC:** Ensure all conditions are met to achieve a Zero DTC environment.
- **Execution Ready:** With the Zero DTC environment set, proceed to the next phase of execution.

Responsibilities:Test Case Design and Execution For Failsafe

Requirement Analysis:

- Review the Master Requirement Sheet.
- Filter requirements based on the vehicle type (HEV, EV, or Normal).

Test Case Design:

- Design test cases based on the filtered requirements.
- Specify unique preconditions for each requirement to ensure accurate testing.

Automation Scripting with VtestStudio:

- Utilize VtestStudio to craft automation test scripts tailored for CANoe execution.

Responsibilities:Test Case Design and Execution For Failsafe

Script Integration into CANoe:

- Import the automated test scripts into CANoe, preparing them for execution.

Test Execution:

- Run the test scripts in CANoe.
- Dynamically control parameters such as voltage, variant code, vehicle speed, etc., to simulate various scenarios.

Example:Camera Blockage Test Case Execution

Setting Pre-conditions:

- Ensure the engine is on.
- Set the variant code as per the requirement.

Initiating Camera/Radar Blockage:

- Simulate a blockage by assigning a specific value to the camera or radar signal.

Example:Camera Blockage Test Case Execution

DTC Logging and Verification:

- Verify that the DTC logs with the desired status mask (89 - active state).
- Measure the time taken for detection.

Error Recovery:

- Remove the blockage by assigning the recovery value to the signal.
- Monitor for the status change from active (89) to history state (08).

Example:Camera Blockage Test Case Execution

Execution Platforms:

- We can test the scenario either in the Graphical window to check signal values or using test scripts in VtestStudio.

Flag Condition Verification:

- For test cases without a DTC number, verify the change in flag conditions to ensure requirements are met.

Responsibilities:Test Report Compilation and Review

- **Report Collection:**Gather all generated reports and attach them to the respective test cases.
- **Self-Review:**Conduct a thorough self-review before submitting for the final review.
- **Addressing Comments:**If any comments or discrepancies are found in the report, re-test and regenerate the report as required.
- **Final Review and Publishing:**Once the final review is complete, publish the report in Jira.

Include data such as the total number of pass test cases, failed test cases, and non-applicable tests in the final report.

Conclusion:Challenges Faced During System Testing

ECU Hardware Limitations:

Mismatch between the customer's advanced hardware versions and our available versions can pose challenges. Creating a similar testing environment with a lower version of hardware might not always yield accurate results.

Flashing Risks:

During the flashing process, issues such as power interruptions or flashing the wrong file can render the ECU unusable or "bricked."

Complex Test Environment Setup:

Ensuring all components, like JTAG connectors, Trace32, and harness cables, are in proper condition can be intricate.

Conclusion:Challenges Faced During System Testing

Ensuring Zero DTC Environments:

Achieving a Zero DTC environment and ensuring no false positives or negatives during DTC checks.

Large Volume of Test Cases:

Managing and executing a vast number of test cases, especially when considering different vehicle types and conditions.

Time Constraints:

Meeting tight deadlines while ensuring comprehensive testing can be challenging.

Intermittent Failures:

Some issues might not be consistently reproducible, making them harder to diagnose and fix.

Thank you