AIVA - DLT Analysis Plugin

Ayoub Ghoula - Khaled Kammoun

Table of contents

01

What We Learned During the Internship

Formation & Workshops

Basic Knowledge Overview

ECU Types and Their Roles

Automotive Ethernet Workshop

Understand the Tools Used for Testing

ECU modes and System Behavior

02

Achievements

1. Documentation: Hard Reset Lifecycle

- A complete power cycle or reboot of an ECU that fully restarts its applications.
- The ECU Hard Reset lifecycle starts with a diagnostic request 11_01

ECU Targetstates belong multiplet Function Groups (FG) which are independent:

- **Driving | SRR | MRR | IKS | CustomerApplications | BaseApplications : Off->Startup->On**
- **Data: Off → Startup-> Full**
- **MachineFG: Startup->StartupStep2->running.**
- **Engineering: Off ->Engineering Mode**
- **CountryVariant: Off->DrivingRestOfWorld**

Common Issues in ECU Hard Reset

Issue : Application crashed → (Crash During Software Update - PID[319584])

- **Detection & Hints:**
  - **Filtered DLT logs for ECUID:** Payload: (rx, | tx, | #CUSTOM4) , Apid: (INT | EM | LSM | LOGC) .
  - **Steps:**
    i. CUSTOM4 error detected, indicating an application crash and triggering coredump collection.
    ii. Aurix delays the ISOC reboot by 50 seconds to allow coredump writing.
    iii. During the delay, Aurix disables the performance switch to prioritize coredump collection.
    iv. After the 50-second delay, Aurix forces an ISOC reboot to complete the hard reset.
    v. The system creates a compressed core file (e.g., .gz) and transfers it to /persistent/coredumps/ for debugging.

Common Issues in ECU Hard Reset

Issue : Application crashed → (ECU failed to set **machineFG** to running state due to a **Bs application crash**)

- **Detection & Hints:**
  - **Filtered DLT logs for ECUID:** Payload: (rx, | tx, | #CUSTOM4) , Apid: (INT | EM | LSM | LOGC) .
  - **Steps:**
    i. ECU initiates a hard reset to transition machineFG to the running state.
    ii. System sets the machineFG state to running and awaits confirmation.
    iii. Observed an initial error in setting machineFG to the running state, indicating a transition failure.
    iv. Noted LSM retrying the operation, with errors persisting on attempts 2/3 and 3/3.
    v. Identified a CrashReport in the logs, revealing a Bs application crash as the root cause.

2. Documentation: Flash Mode

- A special operating state of an ECU that allows its software to be flashed. During Flash Mode, the ECU prioritizes the update process, often restricting other operations to ensure the update completes without interruptions.
- ECU Targetstates belong multiplet Function Groups (FG) which are independent in activating and deactivating the Flash mode:

1. Activating the Flash mode using **31 01 0F 0C 03**

- **MachineFG: PlatformOnly**
- **Driving : Flashing**
- **SRR, MRR, IKS, CustomerApplications, BaseApplications, Data: Off**

2. Deactivating the Flash mode using **31 01 0F 0C 00**

- **MachineFG: PlatformOnly -> startup**
- **Driving: Off->Startup->On**
- **DrivingRestOfWorld : CountryVariant**
- **SRR, MRR, IKS, CustomerApplications, BaseApplications, Data : Off->Startup->On**

Common Issues in ECU Flash Mode

Issue 1: ISoc Minion setPowerMode Failure

- **Detection & Hints:**
  - **Filtered DLT logs for ECUID:** ApplicationId (), Payload (rx, | tx, | #CUSTOM4)

  - **Steps:**
    i. ECU initiates a Flash Mode request using UDS service 31_01.
    ii. ECU sends a Flash Mode message to ISOC Minion and awaits acknowledgment.
    iii. ISOC Minion responds with an NRC, indicating a failure to enter Flash Mode.
    iv. ECU retries the request, receiving a second NRC after 5 seconds.
    v. ECU calls setPowerMode(1, 25) on ISOC Minion to enable Flash Mode, expecting a success or NRC response.
    vi. If successful, Flash Mode is activated, and the ECU restricts other operations to complete the update.

Common Issues in ECU Flash Mode

Issue 2: Cv-Soc Minion setPowerMode Failure

- **Detection & Hints:**
  - **Filtered DLT logs for ECUID:** ApplicationId (), Payload (rx, | tx, | #CUSTOM4)

  - **Steps:**
    i. ECU initiates a Flash Mode exit request after a successful update.
    ii. ECU calls setPowerMode(1, 25) on ISOC Minion to confirm Flash Mode activation, expecting a success response.
    iii. ECU calls setPowerMode(0, 0) on CVMinion to exit Flash Mode and resume normal operation.
    iv. CVMinion acknowledges the request, confirming the mode transition.
    v. If successful, the ECU resumes normal operation with updated firmware.

Common Issues in ECU Flash Mode

Issue 3: Application crashed → (XNM application crashed during shutdown in Flash Mode, preventing crash report generation)

- **Detection & Hints:**
  - **Filtered DLT logs for ECUID:** ApplicationId (), Payload (rx, | tx, | #CUSTOM4)

  - **Steps:**
    i. Observed the ECU in startup state after a reset, indicating initialization.
    ii. Detected Crash Reporter (CR) logging an XNM application crash during shutdown.
    iii. Noted the absence of a crash report (e.g., no .gz file) due to terminated support services.
    iv. Identified the XNM application termination via SIGKILL signal as the final event.

3. Documentation: Network Manager (NM)

- **What is NM?**
  - NM (Network Manager) is a message sent from the IPB (main ECU) to other ECUs in the vehicle.
  - Communicates via an Ethernet UDP frame, transmitted every 100ms.
  - Transfers the current states of all ECUs in the network.

- **PNC (Partial Network Communication)**
  1. **PNC = 0:**
     - No NM message sent from IPB.
     - Results in automatic shutdown
  1. **PNC = 159:**
     - every ECU is fully awake

Common Issues Related to the NM

Issue 1: Network Manager (NM) communication loss triggered an aborted ISOC shutdown.

- **Detection & Hints:**
  - **Filtered DLT logs for ECUID:** Apid: INT,Ctid:LCM,Payload : Pnc .
  - **Steps:**
    i. Filtered DLT logs for Apid: INT, Payload: PNC, and observed a PNC value of 0, indicating no partial network communication and NM failure.
    ii. Used Wireshark to identify a communication gap, with a delta > 2 seconds (4:16:18 to 4:16:21), confirming the NM drop on the network.
    iii. Detected error log "ERROR: ISOC shutdown abort – COM!" during the no-communication window, showing the shutdown process aborted due to communication failure.
    iv. Confirmed the state transition from DEGHNDLR ISOC FSM to PREPARE_SHUTDOWN, reflecting a controlled shutdown attempt due to NM loss.

03

Current Progress

DLT Analyzer

Our Custom Python Tool for Log Analysis

What is DLT Analyzer Tool?

- A modular Python tool to simplify and automate the analysis of .dlt files used in the automotive world

---

What We've Built So Far ?

- **Reads raw .dlt files using DltFileReader from pydlt.**
- **Adjusts timestamps, ensuring accurate time representation.**
- **Parses fields like index, time, ecuid, Apid, Ctid, Type, and Payload, cleaning payloads with clean_payload for consistency.**
- **Cleans messages by:**
  - Skipping empty or non-verbose messages (e.g., checks for "non-verbose" in payload or message type).
  - Handling corrupted messages with a safe reader loop, skipping invalid entries.
- **Applies custom filter rules extracted from:**
  - past debugging sessions.
  - documentation.

Conclusion
& Next Steps

Conclusion

- Gained hands-on experience in automotive diagnostics via DLT log analysis
- Documented debugging process, improving ECU behavior visibility

Next Steps

- Continue developing our Python-based tools Documented debugging process, improving ECU behavior visibility
- Implement more advanced filtering, reporting, and error classification features
- Explore integrating AI models to automate anomaly detection in logs

Thank You!

for your attention and support throughout our internship journey.