The Ultimate Truck Hacking Platform
Hardware Testing and Verification Manual

Colorado State University

October 11, 2024

Dr. Jeremy Daily
Spencer Beer
Carson Green

Summary

This document is a testing plan for the Ultimate Truck Hacking Platform (UTHP) hardware design. It includes:

- **Compound Functional Tests**: a concise set of tests that demonstrates the operation of all required features of the UTHP hardware.
- **Commissioning Checklist**: A systematic checklist for recording the results of the tests for each hardware unit by serial number. This allows for effective tracking and quality assurance of individual units throughout the commissioning process.

The outline for the testing document is as follows:

1. Overview of UTHP Hardware
2. Testing Methodology
3. List of Compound Functional Tests
4. Commissioning Checklist
5. Data Recording and Reporting
6. Quality Assurance Protocols
7. References

Overview of UTHP Hardware

The UTHP device, developed for the National Motor Freight Traffic Association (NMFTA) is a multi-tool used for cybersecurity research and testing by connecting to and analyzing the various vehicle networks found within the automotive industry. The UTHP integrates the BeagleBone Black as the main processing unit, connected to various inputs and outputs, transceivers, and communication interfaces such as the LIN and CAN buses.

The key components are as follows:

- BeagleBone Black (BBB): The UTHP cape, the daughter board for the BBB, interfaces through header pins allowing for connections to power, SPI, UART, and other peripherals.
- Supercapacitor: This allows the device to be unplugged from a power source safely, to maintain the image on the BBB.
- BitMagic Logic Analyzer (LA): The BitMagic is used as a LA to view live data transported on various buses.

- CAN transceivers: Four CAN transceivers are found on the UTHP, two supporting CAN FD.
- External Connectors: A Deutsch-9 pin connector (DB9), DSUB-15 connector, and banana jacks are externally facing allowing a user to choose between which interface to use in order to connect to a network.
- Buffers: Included to assist with safe and reliable signal transmission between components and ensure protection against voltage mismatches.
- Mikroe-Click: Provides extensibility, enabling the addition of new modules if desired.
- Voltage Regulation: Creates a reliable 5V DC output from +12V DC.
- SSC P485 Breakout Board: Allows for transmitting and receiving PowerLine Communication (PLC) signals between the BBB PRU and a PLC line.
- Real Time Clock (RTC): Keeps time when the UTHP is shut off.
- LED Indicators: Used for debugging and status indication.
- J1708: Interfacing for the J1708 (coupled with PLC for one circuit).
- LIN: LIN bus driver is included for additional communication / analysis purposes.
- UART / SPI / I2C: Used between PCB components or between the UTHP and computer for data transmission.

Testing Methodology

The testing methodology for the UTHP hardware focuses on efficiently verifying all essential components and functions through a series of compound tests. The goal is to ensure that the platform's critical features operate as expected under normal conditions. The tools used are a computer / personal laptop, multimeter, USB cable, and access to the various vehicle networks that the UTHP includes functionality for. All results will be recorded in the commissioning checklist, with detailed logs for any failures. This approach ensures comprehensive coverage of all critical systems while maintaining efficiency through the use of compound tests. The testing procedures and expected outputs are seen in the following sections - **List of Compound Functional Tests** and the **Commissioning Checklist.**

List of Compound Functional Tests

1. **Power-On Self Test (POST)**
    - **Objective**: Verify system power stability and initial hardware diagnostics.
    - **Procedure**: Power on the UTHP hardware and monitor the power rails (3.3V, 5V) using a multimeter or oscilloscope. Confirm that all LEDs indicate proper status (e.g., power, system boot, fault indicators).
    - **Expected Results**: All power rails should be within specification, and LEDs should show the correct startup status.
2. **CAN Communication Test**
    - **Objective**: Test the functionality of all CAN channels (CAN0, CAN1, CAN2, CAN3) in a single test.
    - **Procedure**: Set up a CAN network using a CAN analyzer and exchange messages across all CAN channels while monitoring for signal integrity.
    - **Expected Results**: Successful transmission and reception of CAN messages without errors.
3. **Power Management and Safe Shutdown Test**
    - **Objective**: Validate the power management system, including the operation of the supercapacitor during power loss.
    - **Procedure**: Simulate a power outage by disconnecting the primary power supply. Monitor the system to ensure it remains operational long enough for a safe shutdown, with the supercapacitor providing backup power. Confirm that the system shuts down gracefully without data loss.
    - **Expected Results**: The system should remain powered long enough for a safe shutdown when the primary power is lost, and data should be preserved.
4. **UART Test**
    - **Objective**: Ensure correct data transmission and reception across UART5 and UART4 interfaces.
    - **Procedure**: Connect UART5 and UART4 to an external serial monitor or loopback connector. Send data packets through both interfaces and verify that the

data is correctly transmitted and received without errors.
    - **Expected Results**: Both UART5 and UART4 should transmit and receive data accurately, without loss, corruption, or delays. All data packets should match the sent and received data.
5. **BitMagic Test**
    - **Objective**: Verify proper operation of the BitMagic interface for logic analysis and debugging.
    - **Procedure**: Connect BitMagic to the UTHP's BitMagic connector, while connected to CAN and J1708. Open PulseView on a laptop and verify data lines are toggled when data is received,
    - **Expected Results**: BitMagic should accurately capture and display the state changes of the pins, with no errors or missed transitions.
6. **Connector Tests**
    - **Objective**: Confirm functionality of all connectors under normal operation.
    - **Procedure**: Connect the DSUB-15, DB9, and banana jacks to their corresponding networks (one at a time) and run their software equivalent to ensure the connectors have no shorts or wiring issues.
    - **Expected Results**: All connectors should allow for data transmission and reception, or for power/ground wires should contain the proper voltages. No shorts or mismatched wiring present.
7. **PLC / J1708 Test**
    - **Objective**: Validate the Power Line Carrier (PLC - J2497) and J1708 communication protocols.
    - **Procedure**: Connect a J1708-compatible device to the UTHP and initiate communication over the J1708 bus. Simulate PLC communication using the appropriate signals and measure performance using a PLC tester or J1708 analyzer.
    - **Expected Results**: Both J1708 and PLC signals should be correctly transmitted and received without error. The system should handle simultaneous operation of both protocols.
8. **LIN Test**

- **Objective:** Verify LIN communication between UTHP and LIN-capable devices.
- **Procedure:** Connect a LIN slave device to the LIN bus. Send data commands from the UTHP to the LIN device and confirm the accuracy of the received data using a LIN protocol analyzer.
- **Expected Results:** LIN communication should be free from errors, with accurate transmission and reception of data. The bus should handle multiple LIN devices without conflicts.

9. **RTC Test**
    - **Objective:** Ensure the Real-Time Clock (RTC) functions properly and maintains time.
    - **Procedure:** Set the RTC to a known time and let it run for a predetermined period. Power cycle the system, then check the RTC to confirm that it maintains the correct time after the reset.
    - **Expected Results:** The RTC should retain the accurate time across system resets and power cycles, without drifting or losing track of time.

Commissioning Checklist

Tester Name: _____

Unit Serial Number: _____

Hardware Version/Revision: _____

Firmware Version: _____

Test	Test Date	Result [Pass/Fail]	Remarks / Issues
1. POST			
2. CAN			
3. Safe Shutdown			
4. UART			

5. Bitmagic			
6. Connectors			
7. PLC / J1708			
8. LIN			
9. RTC			
Other: _____			

Tester Sign-Off: _____

Data Recording and Reporting

Data Collection Process: All test data should be recorded in real-time during testing to avoid data loss or inaccuracies. Each test result, including pass/fail outcomes, test parameters, and any observed issues, must be documented in the **Commissioning Checklist** or a digital database.

Test results should include the following minimum details:

- Serial number of the hardware unit.
- Date and time of the test.
- Name of the tester or responsible personnel.
- Firmware and hardware versions.
- Test conditions (e.g., temperature, load).

Data Format and Storage:

- **Format**: All test results should follow a standardized format, including consistent data fields (e.g., serial number, test name, result, remarks). Ensure all testers are trained to fill out the forms uniformly.
- **Storage**: Test data should be filled out in the **Commissioning Checklist**, and attached to the corresponding serial-numbered device upon delivery.

Failure and Issue Reporting: If a test fails or reveals an issue, the failure mode and cause should be thoroughly documented in the **Remarks** or **Failure Mode** fields. Additional steps may include attaching diagnostic data (e.g., CAN bus logs, oscilloscope traces) to the test report for further analysis. Contact Carson Green if any errors are uncovered with the hardware. Failed tests should trigger a **Failure Report**, summarizing the nature of the issue, the specific test affected, and a proposed plan for resolution, which may include repair, retesting, or redesign of the hardware.

Audit Trail and Version Control: To maintain an accurate history of test results, every test record should include a timestamp and the identity of the person who performed the test. This

audit trail ensures traceability and accountability for every stage of the testing process. Use version control to track changes in test results, especially when hardware or firmware versions are updated. Each time a test is rerun, ensure the results are saved as a new version, preserving the history of earlier test outcomes.

Quality Assurance Protocols

- Each test in the commissioning process must have clearly defined **pass/fail** criteria to ensure objective, consistent evaluation of hardware performance.
- Periodic **test validation** should be performed by rerunning known-good test cases to ensure the testing setup remains accurate and reliable. This helps identify issues related to testing equipment or procedures.
- If a unit fails a test, a structured re-test process must be followed to diagnose the root cause of the failure. The re-test should:
    - Investigate whether the issue is reproducible under similar conditions.
    - Involve a deeper analysis of the failed subsystem (e.g., communication logs, power trace analysis).
    - Capture diagnostic data to support further investigation (e.g., logs from a CAN analyzer, signal waveforms).
- Each hardware unit should be traceable through its serial number, and all test results associated with that serial number must be easily accessible in case of future inquiries, audits, or customer feedback.
- The quality assurance process should incorporate **continuous improvement** mechanisms, allowing testers and engineers to suggest improvements to the test procedures based on their experience.
- Each unit tested must receive formal sign-off. The sign-off should confirm that:
    - All required tests were completed.
    - The unit met all acceptance criteria or was successfully re-tested and passed.
    - The data was recorded and logged accurately.

- Only after this sign-off should the unit be approved for deployment or customer delivery.

References

1. https://github.com/SystemsCyber/meta-uthp
2. https://github.com/SystemsCyber/UTHP