The Ultimate Truck Hacking Platform Hardware/Software Testing and Verification Manual

October 11, 2024

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.

Overview of UTHP Hardware

The UTHP device 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. **Flash The UTHP: (room 342)**
   a. Remove the BeagleBone carefully from the green UTHP main board.
   b. Use https://github.com/SystemsCyber/UTHP/releases and download the latest release.
   c. Use Balena Etcher to flash the downloaded file (from above) onto a MicroSD card with at least 4GB of storage. Select "Flash from File".
   d. After Balena Etcher is finished, insert SD card into BeagleBone. Make sure USB cable is not connected at this point (no power to the board).
   e. Insert USB-Mini cable to power on the BeagleBone.
   f. Using Putty or a terminal, ssh into the BeagleBone using "ssh root@192.168.7.2" and enter the password for the pre-production image "ultimate".
      i. NOTE: If the above login does not work, try username "uthp" and password "UTHP-R1-XXXX".
   g. Plug in an ethernet cable and wait a few seconds.
   h. On the UTHP command line, run "emmc-flasher". This script flashes the board's memory and configures the settings we want. It will reboot, requiring you to log in once more after it is finished. Run "emmc-flasher" again, and answer the prompts: timezone → Denver (3), and enter the unit serial number found on the corresponding UTHP enclosure (Serial number is in the form: UTHP-R1-XXXX)
   i. You can now unplug the device, remove the SD card, plug it back in, then ssh into the device using "ssh uthp@192.168.7.2" with the password being the serial number (UTHP-R1-XXXX).
   j. If the above processes succeed, unplug the device and bring it to the garage bay for the next stage of testing.
2. **Complete Assembly**
   a. You may now attach the screws for the end caps to complete the process. Please also screw on the black

nut onto the Deutsch 9-pin connector until it is hand-tight.

3. **Software Tests**
    a. Follow https://github.com/SystemsCyber/meta-uthp/blob/scarthgap/recipes-devtools/uthp-tests/files/uthp-tests/README.md
4. **Hardware Tests**
    a. Follow https://github.com/SystemsCyber/UTHP/blob/main/Testing/Hardware/UTHPHardwareTesting_Physical.pdf
5. Ready for Production
    a. Login and run “make production-ready”

Commissioning Checklist

Tester Name: ____

Unit Serial Number: ____

Hardware Version/Revision: _____________________________________

Firmware Version: ____

Test	Test Date	Result [Pass/Fail]	Remarks / Issues
1. Hardware			
2. Core-Tests			
3. PLC-tests			
4. Remote-Tests			
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. 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