A REPORT

ON

Firmware verification for Automotive Wireless Battery Monitoring Systems

BY

SAI KARTIK **2020A3PS0435P**

Prepared in partial fulfilment of the Practice School-II Course BITS F412

AT

Analog Devices India

A Practice School – II Station of

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, PILANI CAMPUS

September 2023

Contents

List of Figures

List of Tables

Abbreviations

wBMS	**w**ireless **B**attery **M**onitoring **S**ystem
OEM	**O**riginal **E**quipment **M**anufacturer
BMIC	**B**attery **M**onitoring **I**ntegrated **C**ircuit
WIL	**w**BMS **I**nterface **L**ibrary
SPF	**S**ingle **P**oint **F**ault
LF	**L**atent **F**ault
ASIL	**A**utomotive **S**afety **I**ntegrity **L**evel
SM	**S**afety **M**echanism
GPIO	**G**eneral **P**urpose **I**nput / **O**utput
FI	**F**ault **I**njection

Chapter 1

Introduction

1.1 Motive

There is a growing demand for advanced cockpit electronic systems as a result of the automotive industry's ongoing shift towards electric mobility. These technologies are essential for enabling comprehensive vehicle component monitoring and for providing the occupants with a clear, comprehensive overview of the current state and overall health of the vehicle.
The battery of the car is by far the most important of the crucial factors that need constant monitoring. This is particularly important for battery packs made mostly of Li-ion substrates because of the potential for severe, potentially dangerous consequences if their health and charge states are not strictly monitored.
Given these factors, Analog Devices Inc. provides a wBMS solution that is simple to integrate into the automotive setting. Notably, such technology is finding support with Tier 1 auto suppliers for example Stellantis and Visteon, who work closely with OEMs for example Tata, Honda, and Tesla. End-to-end solutions that enable users to thoroughly monitor various aspects of battery health are finally provided as the result of these collaborations.

1.2 Contribution to the project

This project has provided the chance to significantly advance the implementation and verification of functional safety, particularly at the level of battery monitoring sensors, the exact details of which will be discussed in the later sections.

Chapter 2

Product walkthrough

2.1 Hardware setup

In the overarching architecture of the wBMS, the fundamental structure comprises two primary components: managers and nodes. Notably, a single configuration can accommodate multiple instances of both managers and nodes. Each node encompasses a combination of essential elements, including a radio, microcontroller, and a sensor, referred to as the BMIC. The combination of the radio and the microcontroller is called the Pinnacle.

Conversely, the manager mirrors this fundamental setup, featuring a radio and microcontroller specifically configured to receive data transmitted from the nodes. Facilitating wireless communication, nodes transmit vital information collected from the BMIC to the manager through the radio interface. The manager, in turn, functions as the central hub for collecting and aggregating this transmitted data.

To facilitate seamless utilization of this collected data by other microcontrollers, a specialized software component known as the WIL is employed. This library when used with the applications of the client microcontroller, serves as the pathway through which data from the manager is made accessible and readily deployable for various purposes and applications.

It is important to recognise that the integrity of transmitted data may be vulnerable to numerous errors along its path due to the complex system architecture. Notably, Single Point Faults (SPFs) and Latent Faults (LFs) are two different sorts of faults that could potentially be encountered by the BMIC, the main source of data. Other components in the system can also encounter faults and specific guidelines have been set in place to ensure each component that is involved in the system is rated to the highest standard of ASIL ratings: ASIL-D. However, the focus of this report will be on the BMIC and the scripts that run on the BMIC (BMS Scripts).

SPFs are situations in which at least one system component deviates from conformity with the ASIL-D requirements, which constitutes an instant and abrupt breach of safety goals. These defects include a wide range of potential problems, from incorrect multiplexer configurations to overvoltage accidents that could damage any cell or the BMIC's GPIO voltage. An extensive list of these potential faults is described in a safety manual created for the BMIC.

The safety guideline also outlines several crucial Safety Mechanisms (SMs) that must be adhered to. These techniques are intended to guarantee that, even in the presence of defects, the system can continue to warn relevant parts about the presence of a fault while supporting the ultimate goal of adhering to ASIL-D requirements.

The Enforcer, a physical component of one of the wBMS solutions offered by ADI, checks the BMS scripts' adherence to SMs. Additionally, this component must adhere to its own rules in order to receive an ASIL-D rating.

2.2 Software setup

Given the multiple pieces of hardware present in this setup, there has to be relevant software that runs on each of them. The software that essentially drives the BMIC is known as the BMS container.

BMS Scripts are assembly-like written codes that instruct the BMIC to gather various data points at various intervals. Since the scripts are essentially text files, they must be compiled into binary files that can be directly read by the BMIC. This process is done by an external compiler. To ensure proper compilation of the scripts, certain functionalities are built within the compiler to make sure the scripts it compiles adhere to decided safety standards.

Once the container is transferred to the node, the Enforcer schedules the tasks and sends the instructions over various time slices in fixed intervals to retrieve the data the manager would ultimately make available.

A methodical approach is used, with scripts serving as a key instrument, to ensure that the BMIC complies with the strict safety regulations. The key component of this strategy is the intentional introduction of flaws into the system from an outside source. The focus then shifts to whether or not the scripts that are being executed produce warnings about these injected errors.

This technique is a vital check on the BMIC's compliance with the stated SMs required for its operation. This verification technique evaluates the BMIC's capacity to recognise and react to deviations from safety protocols by purposefully introducing problems. Thus, it makes a substantial contribution to the overriding objective of making sure that the BMIC complies with the set safety criteria.

Since it is virtually impossible to inject hardware faults (such as internally tampering with the multiplexer lines), we use an external device known as an ”emulator” that simulates the responses given by the BMIC under cases of Fault Injection(FI). The emulator is also designed to work like a regular node in other circumstances. It essentially overrides the communication lines between the Enforcer and the node providing the respective fault data to the enforcer when a fault is injected.

The system can also operate in several ”modes” to synchronise the states of all devices. These modes enable the manager and the node to communicate effectively. Additionally, they also enable the node to continuously gather data.

Chapter 3

Testing methodologies

3.1 Original test scripts

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

3.2 Python packages

3.2.1 Flashing and logging

3.2.2 Explorer APIs

3.2.3 Emulator package

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

Chapter 4

Conclusion and future work

4.1 Conclusion

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

4.2 Future work

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.