OST
Ostschweizer
Fachhochschule

Fleet Monitoring System

Requirements Specification

Florian Baumgartner, Luca Jost

Advisor: Martin Willi
Supervisor: Prof. Beat Stettler

October 6, 2021

Acronyms

CAN Controller Area Network. 3–5

FMS Fleet Management System. 3–7

GNSS Global Navigation Satellite System. 7

IMU Inertial Measurement Unit. 3–5, 7

IoT Internet of Things. 3

IP Internet Protocol. 3

LAN Local Area Network. 3–5

LED Light-emitting Diode. 4

NAC Network Access Control. 3

PCB Printed Circuit Board. 4

SAE Society of Automotive Engineers. 5

SoC System on a Chip. 2, 4

USB Universal Serial Bus. 3–5

WLAN Wireless LAN. 3–5

Glossary

ESP32 Is a series of low-cost, low-power System on a Chip microcontrollers with integrated Wi-Fi. 4, 5

1 Introduction

Onway AG offers WLAN and network access control solutions and software development. Their main fields of business are solutions for Network Access Control (NAC) as well as communication access for public transport. They are known for developing specialized industrial IoT applications. Onway AG is interested in providing an elegant solution for public transport fleets (e.g. buses) to gather low-level vehicle data and transmit them to a cloud-based system. This information can then be used to monitor the state of the vehicle and inform about possible issues in real time.

2 Task Definition

To collect vehicle data of the Fleet Management System (FMS) and provide IP-based access, a dedicated device has to be developed. It acts as a gateway (communication bridge) and connects the internal CAN-Bus to the on-board network router via LAN or WLAN.
A general block diagram of the system is shown in Figure 2.1.

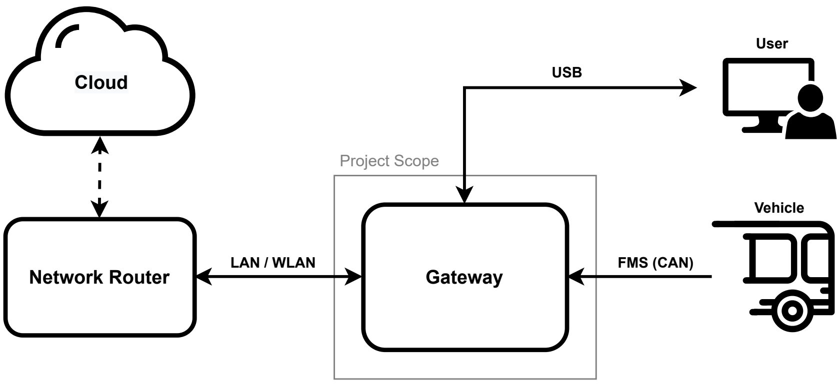


Figure 2.1: *System Block Diagram*

FMS-Packets are received and processed by the gateway. A configurable filter decides which packets get forwarded and limits the maximal transmission update rate. The network router acts as a receiver, which can upload the information over the internet to a cloud-based system. Device settings as well as the filter configuration can be updated by the user without the need of re-uploading a new firmware image. The user can access the device by an USB interface.
To gather additional motion data of the vehicle, an Inertial Measurement Unit (IMU) will be integrated. The collected motion data gets processed and will be available for further usage.

Various use cases were defined in Section 4.

3 Product Requirements

3.1 Hardware

The custom built hardware contains all necessary components integrated on a single Printed Circuit Board (PCB). The hardware needs to comply with the following requirements:

- The hardware shall be based around an ESP32 System on a Chip (SoC).
- The hardware shall use LAN/WLAN to communicate with the network router.

  The Ethernet interface shall be galvanically isolated from the rest of the hardware.
- The hardware shall have a CAN interface to receive FMS-packages from the vehicle.

  The CAN interface shall be galvanically isolated from the rest of the hardware.
- The hardware shall have an Inertial Measurement Unit based on a 3 axis accelerometer.
- The hardware shall be able to show the current device status (e.g. LEDs).
- The hardware shall be capable of operating with 9 to 28 Volts DC.
- The hardware shall be enclosed in a case in order to minimize dust and moisture getting into the system.
- The hardware should have an USB interface for configuration.

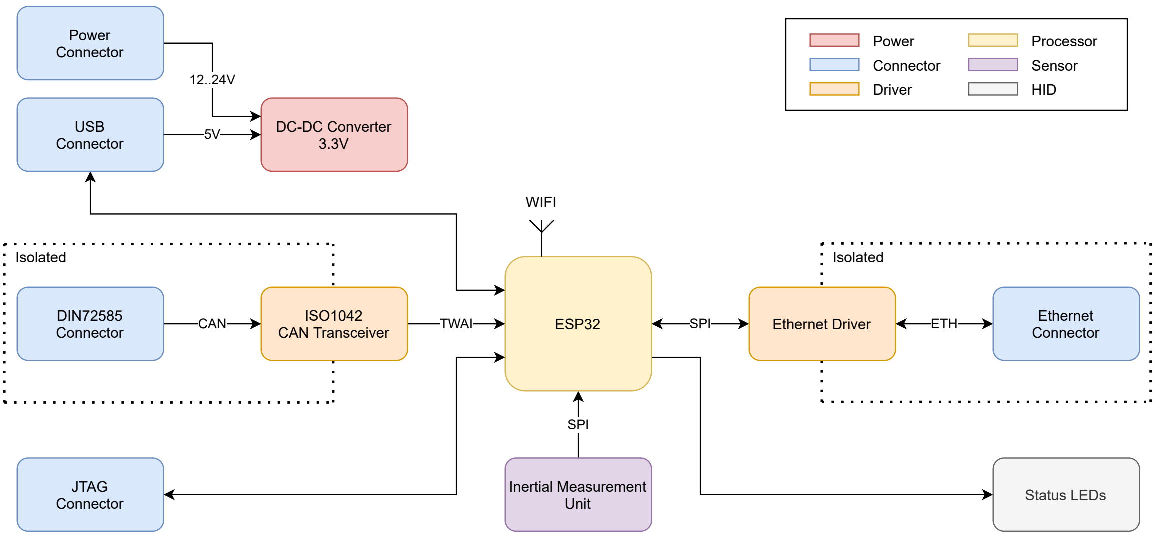


Figure 3.1: *Hardware Block Diagram*

3.2 Protocol and Interface Standards

The following protocol standards will be supported in this project:

- WLAN-Interface: Wi-Fi 802,11b/g/n
- LAN-Interface: 10Base-T / 100Base-TX
- USB-Interface: USB 2.0 (Device)
- CAN-Interface: SAE J1939
- FMS-Protocol: Version 04 (17.09.2021)

3.3 Firmware

The embedded firmware will run on the ESP32 and will be written in C. The requirements for the firmware are as follows:

- The firmware shall read FMS-packages and prepare them for transmission.
- The firmware shall transmit the packages to the router.

  A filter shall be implemented to select the packages to be sent out.

  A filter shall be implemented to select the maximum update rate of a package.

  A filter shall be implemented to send data only on change.
- The firmware shall display the current device status to the user.
- The firmware shall read the IMU-data.
- The firmware shall transmit the IMU-data.
- The firmware should read a configuration file from the router (e.g. the filter of what packages shall be sent).

3.4 Network Router Communication Tool

The network router communication tool is the sole device communicating with the hardware. The system has the following requirements:

- The system shall handle communication between the embedded system and the host.
- The system shall store the streamed data and make it available for further processing.
- The system should host files for the device (e.g. configuration file).

3.5 Configuration File Creator

The configuration file creator helps the user to easily change and adapt the behavior of the FMS-packet transmission. For each FMS command-type, specific filters can be applied (referenced in Section 3.2). The configuration file creator fulfills the following requirements:

- The file creator shall have the function to load and store configuration files.
- The file creator shall provide axes to change the filter settings.

3.6 Graphical Visualizer (optional)

This Python-based tool allows visualization of data that has been gathered and transmitted by the device. It should be implemented as a stand-alone application which primarily can be used offline. The graphical interface helps the user to easily understand the presented data. The requirements for this tool are as follows:

- The graphic visualizer shall open files containing FMS data packets.
- The graphic visualizer shall visualize user selected command-types.

4 Use Cases

4.1 UC1 Driving Quality Analysis

Having a clear understanding of a drivers behavior, is very important for fleet operators. Having data about the vehicle operators can lead to less emissions, improve customer satisfaction and improve the drivers behavior. Our system can be deployed to gather driving data in a very detailed manner. The Inertial Measurement Unit provides accurate motion data of the vehicle, which leads to a better understanding of the driving performance.

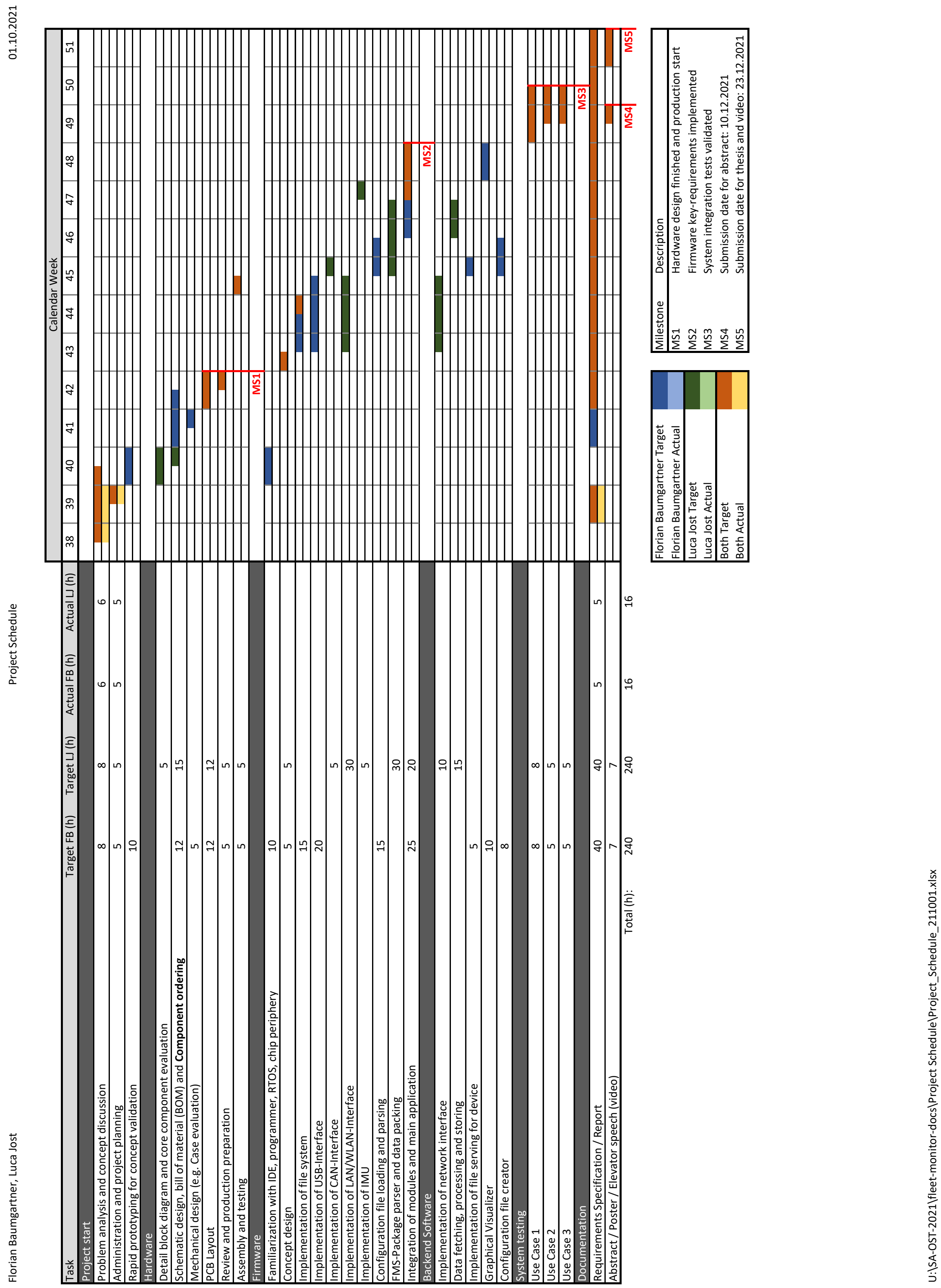
4.2 UC2 Vehicle Maintenance Report

Managing a large fleet of vehicles requires enormous effort. In order to minimize down time it is important to preemptively warn about upcoming and present issues. Our system makes it possible to gather continuous information about the status of the fleet without physical access.

4.3 UC3 Real Time Telemetry Data

Most of modern vehicles are already equipped with GNSS receivers for providing location updates to the Fleet Management System. However it is difficult to differentiate if a vehicle is currently parked or just slowly moving (e.g. due to traffic jams). With the help of knowing the exact driving velocity, a better prediction of potential arrival time can be achieved. Additional data, like the state of passenger doors can directly inform the system about a potential delay of departure.

5 Project Schedule



Florian Baumgartner, Luca Jost — Project Schedule — 01.10.2021

Task	Target FB (h)	Target LJ (h)	Actual FB (h)	Actual LJ (h)
Project start				
Problem analysis and concept discussion	8	8	6	6
Administration and project planning	5	5	5	5
Rapid prototyping for concept validation	10			
Hardware				
Detail block diagram and core component evaluation		5		
Schematic design, bill of material (BOM) and **Component ordering**	12	15		
Mechanical design (e.g. Case evaluation)	5			
PCB Layout	12	12		
Review and production preparation	5	5		
Assembly and testing	5	5		
Firmware				
Familiarization with IDE, programmer, RTOS, chip periphery	10			
Concept design	5	5		
Implementation of file system	15			
Implementation of USB-Interface	20			
Implementation of CAN-Interface		5		
Implementation of LAN/WLAN-Interface		30		
Implementation of IMU		5		
Configuration file loading and parsing	15			
FMS-Package parser and data packing		30		
Integration of modules and main application	25	20		
Backend Software				
Implementation of network interface		10		
Data fetching, processing and storing		15		
Implementation of file serving for device	5			
Graphical Visualizer	10			
Configuration file creator	8			
System testing				
Use Case 1	8	8		
Use Case 2	5	5		
Use Case 3	5	5		
Documentation				
Requirements Specification / Report	40	40	5	5
Abstract / Poster / Elevator speech (video)	7	7		
Total (h):	240	240	16	16

Milestone	Description
MS1	Hardware design finished and production start
MS2	Firmware key-requirements implemented
MS3	System integration tests validated
MS4	Submission date for abstract: 10.12.2021
MS5	Submission date for thesis and video: 23.12.2021

U:\SA-OST-2021\fleet-monitor-docs\Project Schedule\Project_Schedule_211001.xlsx

Figure 5.1: *Project Schedule*