March 27th , 2024

Project No.15

SMART HELMET WITH LIGHT INDICATORS FOR BRAKES & TURNS

Individual Progress Report

by

Jasmehar Kochhar

``

TA: Nithin Shanthini Praveena Purushothaman

Contents

1. Introduction

Motorcycle safety presents a persistent concern, particularly regarding rider visibility amidst varied road conditions. Our project aims to mitigate this issue by implementing a system designed to enhance the visibility of motorcycle riders through the integration of light signals reflected atop their helmets. The challenge confronting motorcyclists on roadways is their visibility to other vehicles, particularly in scenarios where left turn crashes occur due to oncoming traffic's inability to discern their turn intentions—typically indicated by rear-mounted turn signals. Our proposed solution addresses this concern by equipping helmets with lights situated not only at the rear but also on the sides, significantly reducing the likelihood of oversight or misinterpretation by other road users.

By establishing wireless communication between the helmet and the motorcycle's turn and brake signals in real-time, our system ensures seamless coordination. Leveraging a combination of light sensors, microcontrollers featuring Bluetooth modules, and LED lights, our project aims to enhance rider safety and mitigate the prevalent risks associated with motorcycle visibility on roadways.

While individual responsibilities in our group are minimal, our project thrives on collaborative efforts, with team members working in pairs to tackle different aspects of the motorcycle system. This approach makes knowledge sharing easier, ensuring technical challenges are looked through multiple lenses and faster problem-solving. My role specifically is more aligned with programming as of these weeks and, but I have also worked with my teammates on verification of the design.

2 Design

2.1 Block Diagram

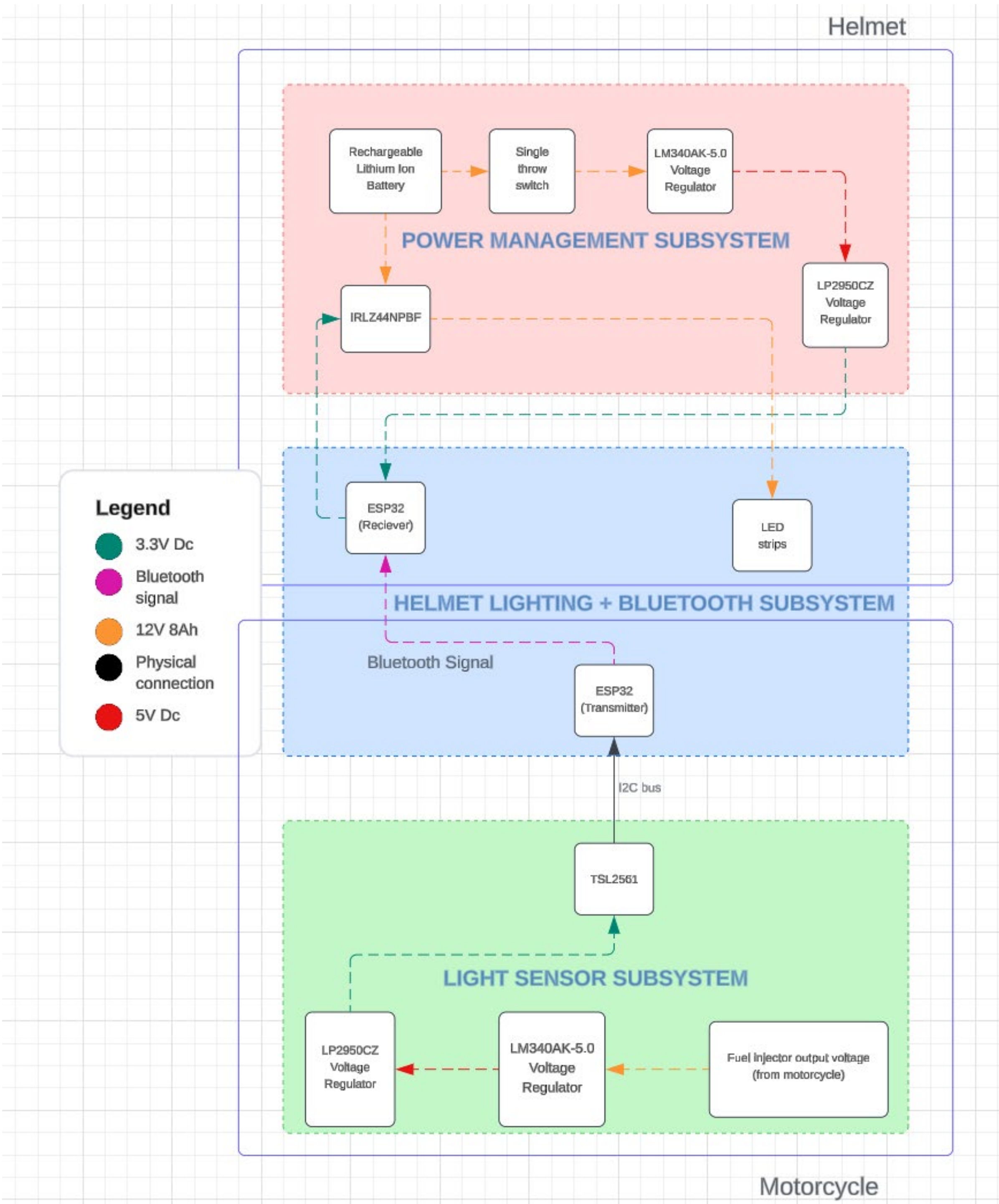

Figure 3: Block Diagram for the lighting system

Our project comprises three primary subsystems: the Power Management Subsystem, the Helmet Lighting & Bluetooth Subsystem, and the Light Sensor Subsystem.

The Power Management Subsystem oversees the provision and regulation of power to essential components such as LEDs and microcontrollers, ensuring stable operation throughout varying conditions.

Additionally, the Light Sensor Subsystem detects the activation of the motorcycle's turn signals and brake lights, relaying this information to the helmet.

My primary focus within the project centered on the Power Management Subsystem and contributed to the development of the Light Sensor Subsystem. While I provided initial concepts for the Light Sensor Subsystem, its implementation was limited due to delayed procurement of necessary lighting components. Nonetheless, my efforts primarily revolved around ensuring efficient power distribution and regulating power for optimal functionality as of now.

2.2 Subsystem Overview

2.2.1 SUBSYSTEM 1: LIGHT SENSOR SUBSYSTEM

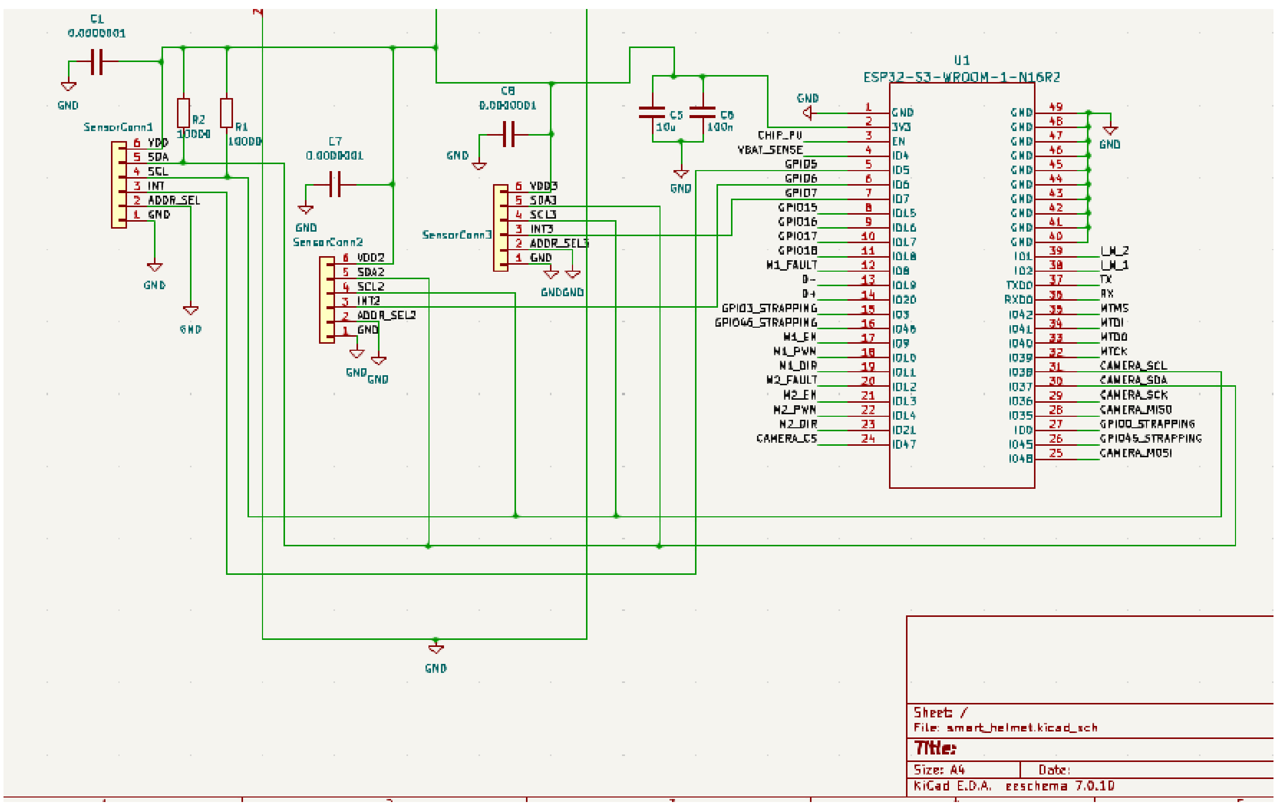

Figure 4: Light Sensor Subsystem **Schematic**

Following the initial design phase, our project encountered its first significant challenge during the testing of PCB #1. It became evident that the boot-up systems we implemented were not functioning as intended due to insufficient pull-up resistors for the boot state. Consequently, a considerable amount of time was spent analyzing and understanding the issue before seeking assistance from another TA, Jason, who had expertise in schematic design for the ESP32 development board. Collaborating with Jason led to a revised schematic, rendering our first PCB obsolete. As a result, we recently revised our PCB board that incorporates improved boot-up systems.

Another problem for the light sensor subsystem on the motorcycle. During the pseudo coding and ESP32 researching phase, we realized the necessity of addressing the sensors differently. Initially, we considered using separate lines for each sensor. However, upon further consideration and consultation with our TA, we learned that utilizing a multiplexer might be a more efficient solution. However, incorporating a premade multiplexer board was not feasible per the requirements of the project. Therefore, we explored the option of altering the sensor addresses themselves, utilizing solder pads for address configuration. This approach maintains compactness and aligns with the project specifications, avoiding unnecessary complexity that could arise from integrating additional ICs onto the board. This decision streamlines the design process, ensuring practicality and adherence to project constraints.

2.2.2 SUBSYSTEM 2: BLUETOOTH SUBSYSTEM - HELMET & MOTORCYCLE COMMUNICATION

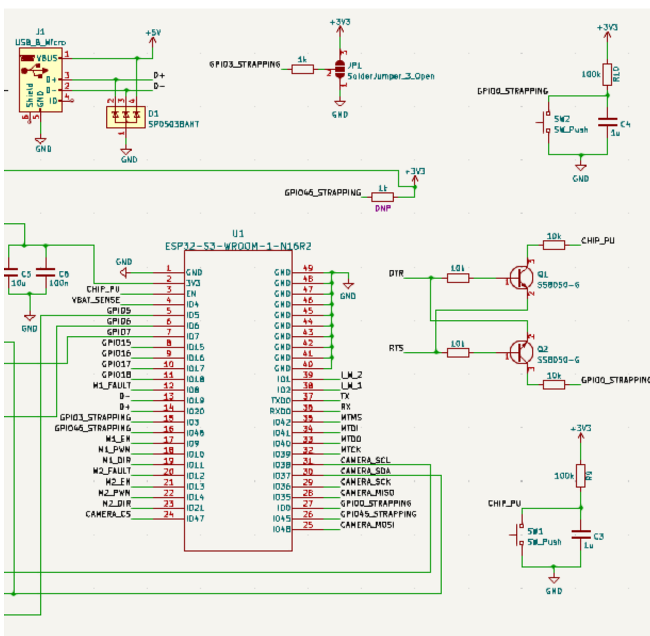

Figure 5: ESP32 with the additional components Schematic

The current focus of our programming work revolves around the ESP32, particularly in setting up interrupt handling and communication between two ESP32 devices. The objective is to receive an interrupt signal from the system, which will then be transmitted to the other ESP32, triggering GPIO pins 5, 6, and 7 to control the LED lights accordingly.

At this stage, extensive testing on the ESP32 hasn't been conducted as we've been primarily occupied with addressing the integration of three different sensor data, which will be resolved in the upcoming round of PCB designs. However, this week marks a crucial phase where we'll be testing the coding, especially focusing on the latency of the system.

Our approach involves leveraging libraries such as the esp32 elf toolchain for efficient development. Initially, we test with one light sensor, periodically obtaining light readings and displaying them on the console.

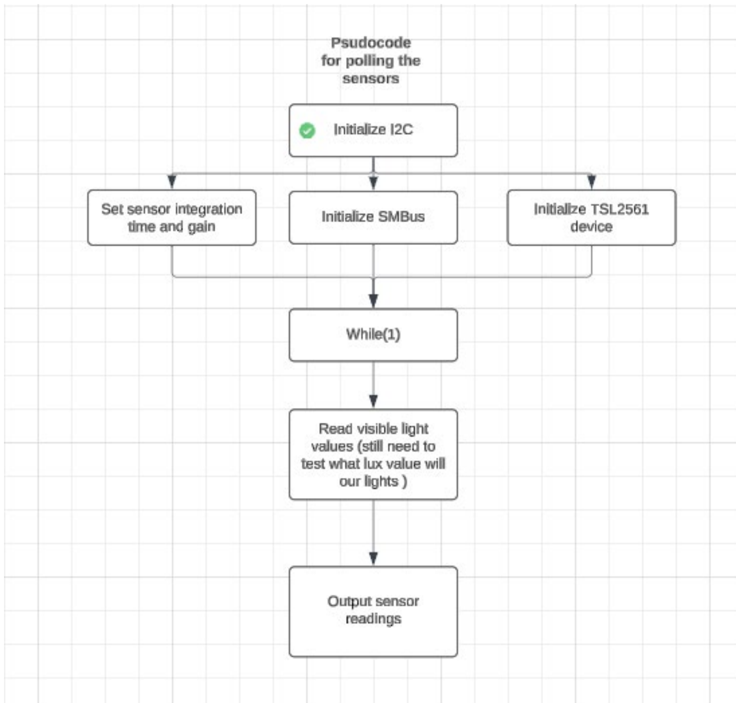

Figure 6: ESP32 current program flowchart

Moving forward, our plan entails implementing interrupt handling to enable the transmission of signals between ESP32 devices for LED control. An essential aspect of this implementation will be verifying the latency, ensuring it remains under 1 second, with a stretch goal of achieving less than 30 milliseconds. One advantage of our design is minimizing code in the polling section and only sending signals upon receiving interrupts, avoiding constant polling and enhancing efficiency.

In summary, our current programming efforts are focused on setting up interrupt handling and communication between ESP32 devices, with upcoming tests aimed at verifying latency and ensuring efficient LED control.

2.2.3 SUBSYSTEM 3: HELMET LIGHTING SUBSYSTEM

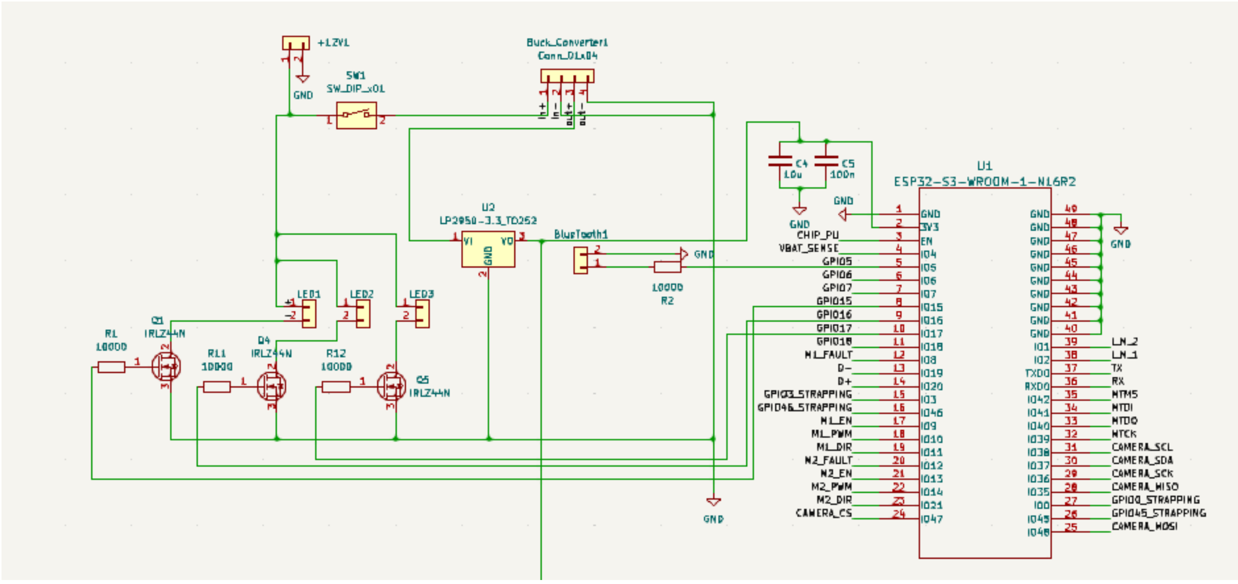

Figure 7: Lighting Subsystem **Schematic**

We have just added extra connections onto the ESP32 so that we can test all three of the lights at once but the major change here is an additional led to show that the Bluetooth connection is on. Other than that, there has been no major change in the system since we haven't had the chance to test the code with the LEDs yet due to us being late in ordering the LED strips.

2.2.4 SUBSYSTEM 4: POWER MANAGEMENT SUBSYSTEM

The decision to switch from linear voltage regulators to buck converters stemmed from issues encountered during testing on the breadboard. The linear voltage regulators proved inadequate for our purposes, as they exhibited shortcomings such as LED fusing or insufficient lighting when the battery voltage fluctuated, particularly under drain conditions. Moreover, the linear regulators generated significant heat and exhibited inefficiency in voltage conversion.

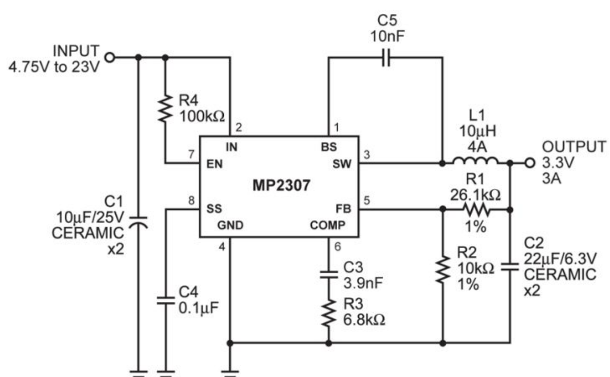

Figure 8: buck converter schematic

Since we haven't yet figured out the resistors needed for conversion here is a general schematic example from the datasheet. We will have to change R1 and R2 in order to obtain our 5 volt output.

Consultation with a TA led to the realization that buck converters would offer superior performance and efficiency compared to linear regulators. However, to maintain a clean output and minimize noise, we opted to retain the linear regulator for stepping down from 5 volts to 3.3 volts, as it provided a more reliable solution for this specific voltage conversion.

Initially, there was confusion regarding the permissible use of external boards, as we had planned to utilize a buck converter board and a multiplexer board sourced from Amazon. Subsequent clarification indicated that the project required integration of the necessary ICs onto our PCB. Consequently, for the upcoming PCB ordering cycle, we plan to incorporate the buck converter IC chip onto the board itself. Additionally, we will address tolerance considerations to ensure optimal performance without relying on external components. This adjustment aligns with project requirements and streamlines the design process by consolidating essential components onto the PCB.

2.3 TESTING AND VERIFICATION

2.3.1 SUBSYSTEM 1: LIGHT SENSOR SUBSYSTEM

Requirements	Verification
1. Sensitivity and Performance requirements: The sensor must distinguish between light conditions and the motorcycle's turn and brake lights. This requires accurate configuration of integration times and gain settings to optimize responsiveness to changes in light intensity.	Once the sensor setup is complete, we will conduct tests in controlled light environments to calibrate its performance. If the interrupts are recorded and printed out in the terminal we will be able to tell that the data was picked up.

2.3.2 SUBSYSTEM 2: BLUETOOTH SUBSYSTEM - HELMET & MOTORCYCLE COMMUNICATION

Requirements	Verification
1. The ESP32 must establish a Bluetooth connection between the helmet and the motorcycle.	We will use a blinking led indicator to signal whether Bluetooth connection was successful or not, as in standard Bluetooth devices for our final product. For initial testing though we will be just sending maybe a print statement to say that it is connected.
2. The Bluetooth subsystem must reliably transfer light sensor data between the helmet and motorcycle.	send the polling LUX values to the other ESP32 to print out to it's terminal and log it then diff check the logs to see if it loses connection
3. The Bluetooth system should be able to remain connected during travel	Same thing as the previous verification. If disconnects implement a way to reconnect

.

2.3.3 SUBSYSTEM 3: HELMET LIGHTING SUBSYSTEM

Requirements	Verification
1. Low latency, not more than 1 second since the light has been turned on or off	An observer will perform a visibility test of the helmet at different angles and in different lighting conditions of the LEDs and time the result

2.3.4 SUBSYSTEM 4: POWER MANAGEMENT SUBSYSTEM

Requirements	Verification
1. Components directly connected with the motorcycle must receive power from the fuel injector voltage output only when the motorcycle is on	Using a multimeter, measure 12 volts output from the fuel injector when the motorcycle is on and 0 volts when the motorcycle is off
2. The system should be able to work under draining power conditions	Test using a battery that is not at full charge and see if the circuit is responding.

4. Conclusion

The delayed purchasing of certain components has temporarily hindered our progress. However, we anticipate regaining momentum this week. Despite this setback, we've utilized the time effectively by continuously testing and refining our circuitry for optimization. With ongoing adjustments and diligent efforts, we remain confident that we'll overcome these challenges smoothly. I believe my workload is quite equal to my teammates since we all do participate in roughly the same amount of work. We should be done with testing Bluetooth between the 2 ESP32 this week for verification for subsystem 1 and 2. Overall code should be finished next week after receiving the LED strips and testing LUX values with them. Then all that is left is testing with the PCB.

References

[1] *https://cdn-shop.adafruit.com/datasheets/TSL2561.pdf*

[2] https://herrmans.eu/news/bike-light-buying-guide-how-many-lumens-or-lux-do-i-need/

[3] https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[4] https://www.link-labs.com/blog/bluetooth-vs-bluetooth-low-energy

[5] https://www.ieee.org/content/dam/ieee-org/ieee/web/org/about/corporate/ieee-code-of-ethics.pdf

[6] https://randomnerdtutorials.com/esp32-pinout-reference-gpios/

[7] https://www.picoauto.com/library/automotive-guided-tests/injector-voltage/ - :~:text=The injector is an electromechanical,is switched via the ECM.

[8] https://www.onsemi.com/pdf/datasheet/lp2950-d.pdf

[9] https://download.datasheets.com/pdfs/2013/4/22/5/11/18/625/txn_/manual/6500lm340-n.pdf

[10]https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf#i2c