Rishikesh Goud Sundaragiri

SMART DRIVE

Functionality of project

The project is composed of two primary functionalities that are intended to be useful for automotive applications. The first function enables the user to regulate the speed of the vehicle by selecting various modes. Some examples of different modes include:

- Eco Mode: This mode is specifically designed to enhance fuel efficiency, which typically involves constraining the vehicle's speed to 50-60% of its maximum speed.
- Normal Mode: This is the default mode that provides a balanced level of performance and fuel efficiency, where the vehicle's speed is approximately 70% of its maximum.
- Sport Mode: This mode is aimed at more dynamic and responsive driving and allows for higher speeds, up to 80-90% of the vehicle's maximum speed.
- Track Mode: This mode is optimized for high-performance driving on a racetrack and typically involves disabling certain safety features and allowing for even more aggressive acceleration, where the vehicle operates at its maximum speed.

The second functionality is the automatic headlights of the vehicle. They switch on when the surrounding light levels become low and switch off during the day or when adequate light is present. However, the user can deactivate the automatic mode and regulate the headlights manually.

Technologies and hardware used in the project

- UART: The UART protocol serves the purpose of receiving user commands via a serial terminal. To distinguish legitimate commands from false ones, a unique set of start and stop characters is assigned to each command, resulting in the formation of a packet. After the packet is received, it undergoes parsing to verify its sender and identify the user-selected mode. For instance, a packet may begin with a '@' start character and end with a '#' stop character. The reception of such a packet would prompt parsing to determine the mode selected by the user. In the event that the packet does not contain the assigned start and stop characters, it is discarded. This mechanism facilitates the differentiation between valid and invalid input commands.
- PWM: The modulation of the pulse width is a technique utilized to regulate the speed of the motor. In the case of the user selecting the sports mode, the speed of the motor is set at 85% of the maximum speed. This same methodology is employed for each of the distinct modes available. By utilizing pulse width modulation, the system can ensure that the motor operates at the desired speed, enabling the vehicle to run optimally in the selected mode.
- ADC: To detect the level of ambient light in the surrounding environment, a Light Dependent Resistor (LDR) sensor is utilized. The sensor provides an analog input signal, which is then directed to the Analog-to-Digital Converter (ADC) pin of the Microcontroller Unit (MCU). The MCU then converts this analog input into a digital value. Depending on the value that is obtained, the onboard LED (which serves as a representation of the vehicle's headlight) is either turned on or off. This mechanism allows for the automatic adjustment of the vehicle's headlights in response to changes in ambient light levels, providing optimal visibility to the driver.

- Touchpad: The system also provides the user with the ability to manually control the vehicle's headlights. By sending a specific command, the automatic control feature is deactivated, and the LDR sensor inputs as well as the ADC module are no longer considered. As a result, the headlights are switched off. The user can then utilize the onboard touch pad, which functions as a switch, to manually turn the headlights on and off as needed. This feature provides greater flexibility to the user and allows them to customize the operation of their vehicle's headlights according to their specific requirements.

In my current project, I intend to explore the ADC module, which hasn't been covered in our assignments yet. Additionally, I plan to delve deeper into the UART protocol and implement secure communication by using unique start and stop characters and parsing incoming packets. While the PWM module has been utilized for controlling the intensity of LED lights in a traffic light assignment, I intend to apply it differently in this project by using it for speed control.

To complete my project, I will need an additional hardware component. As this project is primarily focused on software and obtaining the appropriate motor would be time-consuming, I will not be using an actual motor to demonstrate the speed control functionality. Instead, I will connect the PWM pin to an oscilloscope to display the change in voltage when the user adjusts the speed settings. Since speed and voltage are directly proportional, an increase in voltage on the oscilloscope will indicate an increase in speed. Furthermore, I will utilize a readily available LDR sensor to detect ambient light levels, which can be easily purchased on Amazon without requiring any in-house design work. However, it is important to note that these sensors may not be completely accurate in detecting normal daylight. To address this, I will use a flashlight to demonstrate the presence of light, and turning off the light will indicate that it is night-time. The oscilloscope required for the project is readily available in the lab.

I would like to design a test suite that can test the whole functionality of the project. It will cover all happy, error and corner cases. Mostly, it will be automated or mixture of both manual and automated test cases.

References needed for the project

- KL25Z Reference Manual and Data sheet: https://www.nxp.com/design/development-boards/freedom-development-boards/mcu-boards/freedom-development-platform-for-kinetis-kl14-kl15-kl24-kl25-mcus:FRDM-KL25Z
- To determine the correlation between the amount of light and its corresponding analog values, I would consult the LDR sensor's datasheet.