

CSE438 – Project 3

Our task for Project 3 is to implement an Automotive Back-up Alert System to help drivers of cars know when they are too close to an object.



8x8 LED Grid and Ultrasonic Distance Sensor.

Project Behavior:

Once setup, the system will use an Ultrasonic Sensor to sense the distance between the sensor and an obstacle. It will then display this distance on an 8x8 LED display as a bar. The width of the bar will be proportional to distance (big distance is represented by a wide bar, a short distance represented by a narrow bar). If the distance is below a given safe-distance threshold, the bar should display a flashing “X” as a warning to the driver of the car.

Project Implementation:

1. In User space, implement a library to interface to the 8x8 LED Grid display using the user-space SPI programming interface. The display uses a MAX7219 driver chip. Your interface should allow the display of any pattern of LEDs. More information about the LED display can be found here: http://www.elecrow.com/wiki/index.php?title=LED_matrix_kit
2. In kernel space, implement a library to interface to the HC-SR04 Ultrasonic Distance Sensor. You must use interrupts to interface with the sensor. <http://www.micropik.com/PDF/HCSR04.pdf>
3. In User space, write an application that implements the above described behavior using the 8x8 LED grid and the ultrasonic distance sensor libraries.
 - a. The Safe Distance threshold should be set by default, but modifiable through a command line parameter.
 - b. The display should update continuously, ending only when the user presses “CTRL+C”. The user program must intercept the SIGINT signal and exit cleanly.

BONUS (+40 points):

- Using the spidev.c driver as a template for Kernel-Side SPI communication, implement all of the above functionality as a kernel-space module with the following modifications:
 - a. The safe distance threshold is set by default, but modifiable through an IOCTL call from user space, and a module parameter.
 - b. The display should update continuously, ending only when the module is removed from the system. It should end cleanly. (No need to handle SIGINT).

Submission Information

- Create a working directory to include your source files, makefiles, and readme.
- Comment your source files properly and rewrite the readme file to describe how to use your software. The readme must include a description of necessary electrical connections. Consider submitting a schematic as well.
- Compress the directory into a zip archive file named cse438-lastname-assgn03.zip. Note that any object code or temporary build files should not be included in the submission.
- Submit the zip archive to Blackboard by the due date and time.
- Failure to follow these instructions may cause an annoyed and cranky TA or instructor to deduct points while grading your assignment.

Rubric

Grading Criteria	Points
Kernel Module Reads from Ultrasonic Sensor	20
Use of Interrupts to Read Ultrasonic Sensor	20
User-space library correctly interfaces with LED grid to display information	25
User-space application uses command line to set safe distance variable.	10
User-space application correctly implements project behavior.	25