EE459 Capstone Written Report:

SafeStep - A Fall Detection Device

Team14: Eric Chen, Simon To, Tianhao Wu

Spring 2024

Table of Contents:

I. Understanding Falls

Introducing SafeStep, a pioneering device designed to address a critical and often overlooked issue—fall detection and prevention. Falls represent a significant health hazard, particularly for the elderly, causing severe injuries and even fatalities. The World Health Organization highlights falls as “the second leading cause of unintentional injury deaths worldwide” [1], with older adults facing the highest risk. Falls are not just a major health risk; they also lead to substantial socio-economic burdens. In the United States, “over 30 million falls are reported each year among adults aged 65 and older” [2], emphasizing the scale of this issue. The repercussions of these falls are extensive, ranging from physical injuries that can drastically reduce independence, to significant healthcare costs involving emergency interventions, prolonged hospital stays, and ongoing medical care [2]. The impact of falls extends into legal and insurance domains, particularly in senior living facilities, where they are a source of substantial liability, leading to “legal challenges and increased insurance costs” [3]. Furthermore, the psychological impact of falling includes “fear of falling again, reduced activity levels, social withdrawal, and overall decline in quality of life” [3].

SafeStep utilizes cutting-edge technology to offer an innovative solution for fall detection. The device is equipped with a high-definition MPU-6050 accelerometer and gyroscope all-in-one chip, which measures changes in motion and orientation. This data is analyzed using a proprietary black-box function, derived from a machine-learning linear classifier trained specifically to identify and accurately classify falls. This sophisticated approach ensures that SafeStep provides reliable fall detection, promptly alerting the user and their vicinity, thus facilitating quick medical intervention and potentially mitigating the severity of injuries. In summary, we developed SafeStep to be at the forefront of fall prevention to create a safer environment for at-risk individuals and alleviate the substantial burdens associated with falls.

II. What’s Inside?

Understanding the objectives of SafeStep is key to appreciating the engineering that supports its functionality. This device is engineered to detect falls accurately and initiate timely alerts to

mobilize assistance, potentially reducing the severity of injuries. Each component in the SafeStep system plays a critical role in ensuring reliable performance under real-world conditions. As we transition from the broader goals to the specific technical elements, it's crucial to explore how each part of the system contributes to the device's ability to deliver on its promises. The following section will detail the individual components of SafeStep, emphasizing their integration and functionality in the context of an electrical engineering capstone project.

1. **ATMEGA328P**:
   At the heart of SafeStep is the ATMEGA328P, a single-chip microcontroller that orchestrates our device's graphical user interface (GUI), feedback mechanisms, and fall detection processes. By choosing this chip we are following the requirements of this capstone project to write our main embedded code in C along with the Avrdude package. Additionally, the ATMEGA328P has the exact number of ports we need and its efficient power consumption makes it ideal for a future product that is compact and durable.

2. **MPU-6050 (Accelerometer & Gyroscope)**:
   The MPU-6050 sensor is a 16-bit, 6-axis device that combines an accelerometer and gyroscope to deliver crucial data on acceleration and angular rates to our fall detection algorithm. The sensor's six inputs are essential for distinguishing between everyday activities and potential falls. The gyroscope is particularly useful in scenarios where there is significant movement but no actual fall, such as when someone stumbles but manages to catch themselves. Conversely, the accelerometer is critical in differentiating changes in the person’s orientation but without a rapid descent, such as sleeping.

3. **Electrically Erasable Programmable Read-Only Memory (EEPROM)**:
   SafeStep uses EEPROM for storing data such as the user’s name, weight, and height. By using EEPROM this data is saved between power cycles, so the user won’t have to re-input their biometrics to recalculate the thresholds for fall detection.

4. **Buttons**:
   The buttons on SafeStep enable users to interact with the device, allowing them to

respond to prompts, cancel alarms, and retrieve user information. These interactions are handled as interrupts in the system's code, ensuring that the sensor's detection algorithm can operate in real-time, thus maintaining the device's effectiveness and responsiveness.

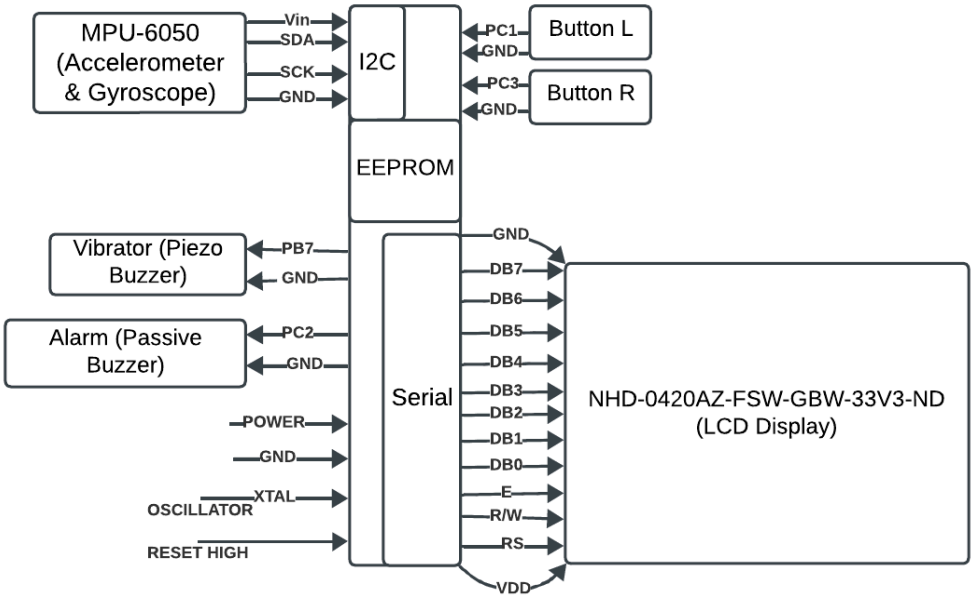
5. **Vibrator (Piezo Buzzer)**:
   SafeStep incorporates a tactile vibrator equipped with a piezo buzzer to notify the user when a fall is detected. The device utilizes Pulse Width Modulation (PWM), controlled by the internal timer of the ATMEGA328P microcontroller, to generate increasingly rapid pulses. This escalation in pulse frequency occurs as the timer assessing the user's consciousness approaches its limit.

6. **Alarm (Passive Buzzer)**:
   SafeStep sounds a passive buzzer as part of its alarm system to alert the surrounding area if the user is deemed unconscious. The sound of the siren is generated through the manipulation of the internal timer on the ATMEGA328P microcontroller, utilizing Pulse Width Modulation (PWM) to create distinct, attention-grabbing siren tones.

7. **NHD-0420AZ-FSW-GBW-33V3-ND (LCD Display)**:
   SafeStep primarily communicates with users through a 2x16 LCD display connected via an 8-bit serial bus, which serves as the central interface for user interaction. This display shows our GUI, any active warnings, and provides options for resetting the device. Additionally, SafeStep features two potentiometers that allow users to adjust the brightness and contrast of the LCD display, enhancing visibility and user comfort according to varying light conditions.

The integration of various components within SafeStep forms a robust system designed to detect falls with high accuracy and provide real-time alerts. By utilizing interrupt-based operations, the system mimics the responsiveness of a real-time operating system (RTOS), which is crucial for delivering timely alerts to caregivers, thereby enhancing safety and providing peace of mind to users and their families. Furthermore, the combination of the MPU-6050 sensor and a tailored machine-learning algorithm allows SafeStep to distinguish between normal activities and actual

falls. This reduces false positives and ensures the system operates reliably under various conditions.



With multiple sensors each using different protocols, there is a great responsibility for us to conduct thorough quality control on each component before proceeding to the final production of SafeStep.

III. Baby Step Testing

Before we began developing the software that integrates all components of SafeStep, we prioritized ensuring each component functioned correctly on its own. Drawing on our experiences from EE109, where we frequently encountered faulty components such as non-responsive buttons, silent buzzers, or inaccurate readings from devices like a potentiometer, we aimed to avoid similar issues in this project. Addressing these component-level issues early on is crucial to prevent minor defects from undermining the overall success of our project.

Here is a list of the software tests we conducted (**note** that the test codes listed will not be included in our final code submission package. We have refined and integrated these test scripts

into functional modules within our main software framework), each accompanied by its goal and reasoning (**note** that all our tests were successful, which assures us that the final product meets the objectives we initially set out in the schematic a section above):

1. **lcd_test.c (using a 8-bit bus)**

   Goal:

   To make sure that each index on the LCD can display all ASCII characters.

   Reasoning:

   As with our experience in EE109, the LCD display is an invaluable tool for debugging within the SafeStep project. It provides real-time debug outputs, allowing us to monitor the device's operation, identify where the setup might be encountering issues, and confirm transitions between different stages of functionality.

2. **button_test.c (using debouncing and interrupt)**

   Goal:

   To make sure that our buttons are correctly debounced and can be recognized as an interrupt.

   Reasoning:

   Proper debouncing of the buttons on SafeStep is critical to ensure user-friendly input and prevent unwanted behavior such as accidental resets. If buttons are not debounced, users may struggle with inputting their information due to multiple readings from a single press, or inadvertently trigger unwanted stage changes due to oscillation.

   The input also needs to be recognized as an interrupt to ensure that SafeStep can offer a real-time fall detection tracking by not getting hampered by delays needed to check button inputs.

3. **buzzer_test.c (using internal timer and PWM)**

   Goal:

To make sure that both piezo and passive buzzer can output a variety of frequencies using a timer and PWM, not delays.

Reasoning:

Making sure that the piezo and passive buzzers can use a timer and PWM (Pulse Width Modulation) rather than delays is essential for several reasons. PWM provides precise control over sound frequency and duration, enhancing the clarity and distinctiveness of alert tones. This method also maintains system efficiency, allowing SafeStep to operate other processes simultaneously without interruptions, crucial for real-time responsiveness in fall detection.

4. **mpu6050_test.c (using I2C)**

Goal:

To ensure that all six axes accurately detect changes in motion, reflecting the correct direction of movement. Additionally, it is crucial that the data is free from corruption to guarantee that it can be reliably interpreted and properly displayed through our LCD display functions.

Reasoning:

We need to ensure that all six axes of the MPU-6050 accurately detect the direction of movement for several reasons.

It is crucial that the data remains free from corruption to ensure the proper initialization of the I2C bus. This includes setting the correct processor speed and BAUD rate for the MPU-6050 to guarantee compatibility. Ensuring the integrity of the data also means it can be accurately stored in variables and displayed clearly on the LCD. This not only enhances the reliability of the SafeStep device but also improves user interaction by providing clear, accurate feedback on the LCD display.

Accurate sensor data is essential for the reliable performance of SafeStep's fall detection feature. Precise readings help differentiate between everyday activities

and falls, preventing false alarms and ensuring the device alerts only when necessary.

With each component now verified to function independently and efficiently, and with the assurance of real-time operation without hard delays, we are ready to advance to the next phase of development. The section below depicts the state machine that will integrate all components into a cohesive system and the rationale behind the GUI we made.

IV. Our Software Blueprint

The C programs that control the fall detection device are organized into distinct filesis based on different functionalities. We provide a concise overview of our implementation to offer a broad comprehension of the software architecture, intending to facilitate future improvements and collaboration.

Software Outline:

- project folder
  - global.h
  - gui.c/h
  - i2c.c/h
  - lcd.c/h
  - main.c/h
  - Makefile
  - mpu6050.c/h
  - mpu6050_res.h

global.h: declare external variables used across files

gui.c/h: graphical user interface that allows users to interact with the device by selecting different menu options, registering user information, and receiving device status.

i2c.c/h: standard initialization code to start communication between mpu6050 and atmega328P through the i2c interface.

lcd.c/h: standard initialization code of lcd that communicates data with atmega328P.

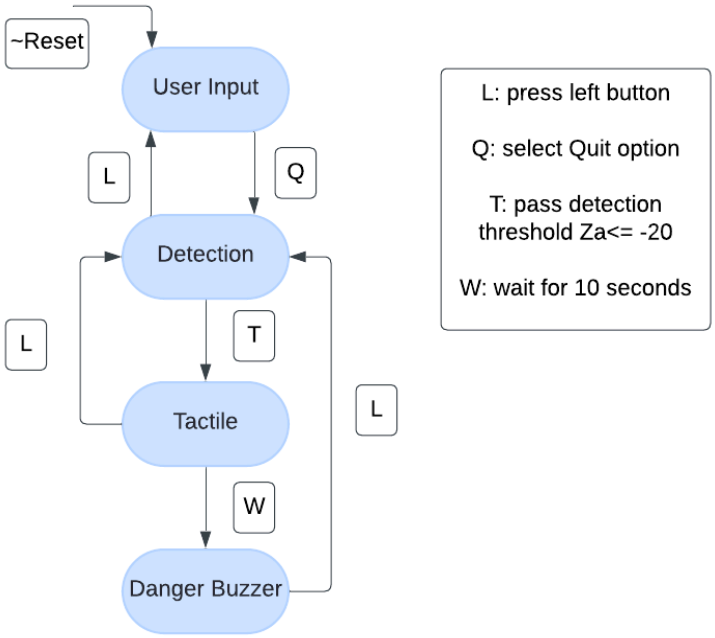
main.c/h: integrate all functionalities in the state machine; implement buzzer and vibrator using atmega328P timers.

mpu6050.c/h: initialization code of accelerometer, gyroscope and temperature sensor; read raw sensor data

mpu6050_res.h: file that defines mpu6050 register addresses

State Diagram:

The following state diagram describes the behavior of our system. For the users to learn to interact with the device, a more detailed description with figures is given in the next section.



Detection Algorithm:

We conducted numerous experiments to determine a detection threshold by simulating various movements, including walking, running, and falling. We observed that the gyroscope fails to form a good metric of detection. Intuitively, angular velocity changes dramatically in many movements other than a fall. On the other hand, acceleration on the z-axis is a better indicator. To distinguish a fall from other movements that produce minor acceleration in the z-axis, we use Za <= -20 as the threshold to focus on negative acceleration towards the ground.

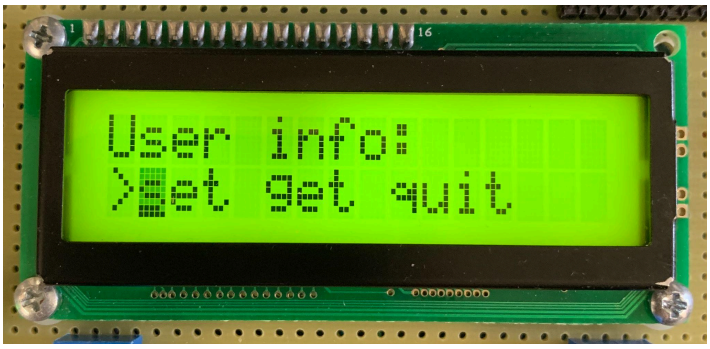
V. A Walkthrough Experience

Stage1: User Input

When the system starts, the LCD screen displays three options:

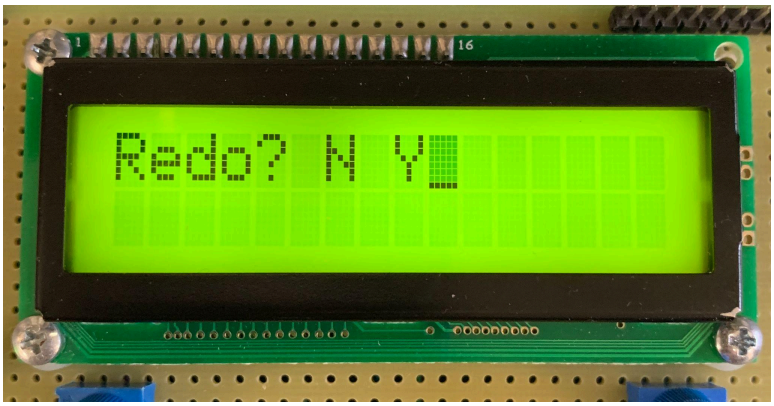
- Set: Enter user information
- Get: Retrieve existing user profiles
- Quit: End this process and start fall detection

By pressing the white button, the user can navigate between different options as indicated by the ">" symbol.



If the user selects "Set", the LCD will first display a series of welcome messages. Next, the user will be asked to enter personal information including name, age, height and weight. Once prompted with "Your name?", press the white button to iterate through alphabetic characters. Then press the black button to select that character. After entering the first name, press the white button once and a "$" sign will appear. Press the black button to finish this part.

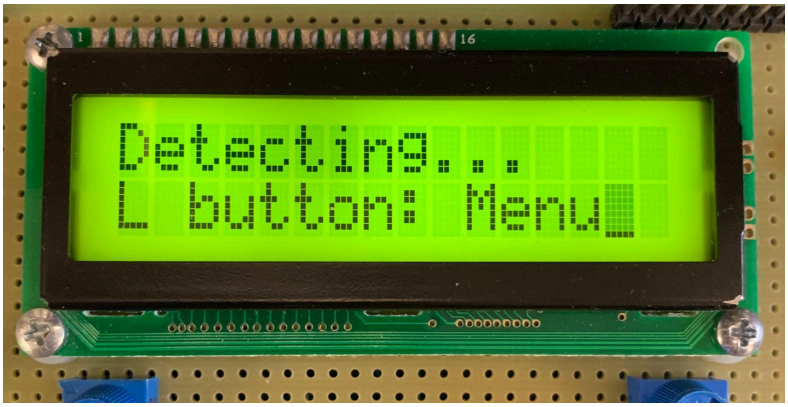
If the user wants to redo it, press the white button for “Y” or “Yes”. Otherwise, press the black button for “No”. Similarly, enter all necessary information and make sure it’s accurate.



If the user selects “get”, the LCD screen displays saved user information automatically.
If the user selects “quit”, the device goes into the detection stage.

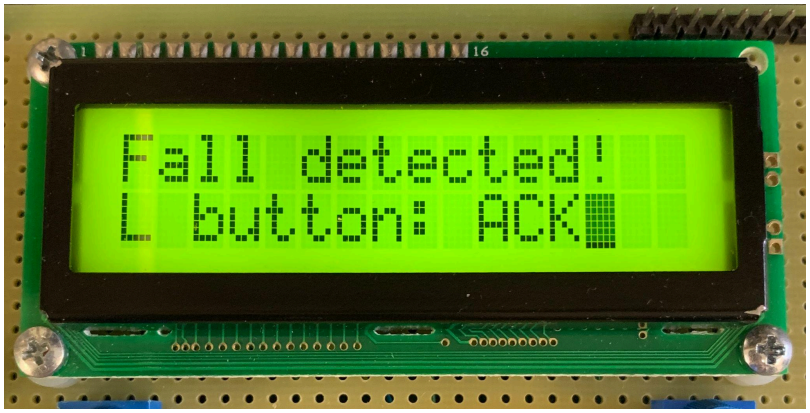
Stage2: Detection

The device initializes the sensors and starts fall detection actively, as indicated on the screen. Press the black/(L) button to navigate to the menu page. Otherwise, if a fall is detected, the device goes to the tactile stage.



Stage3: Tactile

The vibrator starts working. If the user is safe and believes that it's a false alarm, provide an acknowledgment by pressing the black button, then the device goes back to the detection stage. If the user doesn't provide an acknowledgment in about 10 seconds, the device goes to the danger buzzer stage. The vibration stops.



Stage4: Danger Buzzer

The buzzer starts making a loud sound to attract people's attention to the injured user. It keeps buzzing until someone presses the black button to acknowledge the fall. The device then goes back to the detection stage.

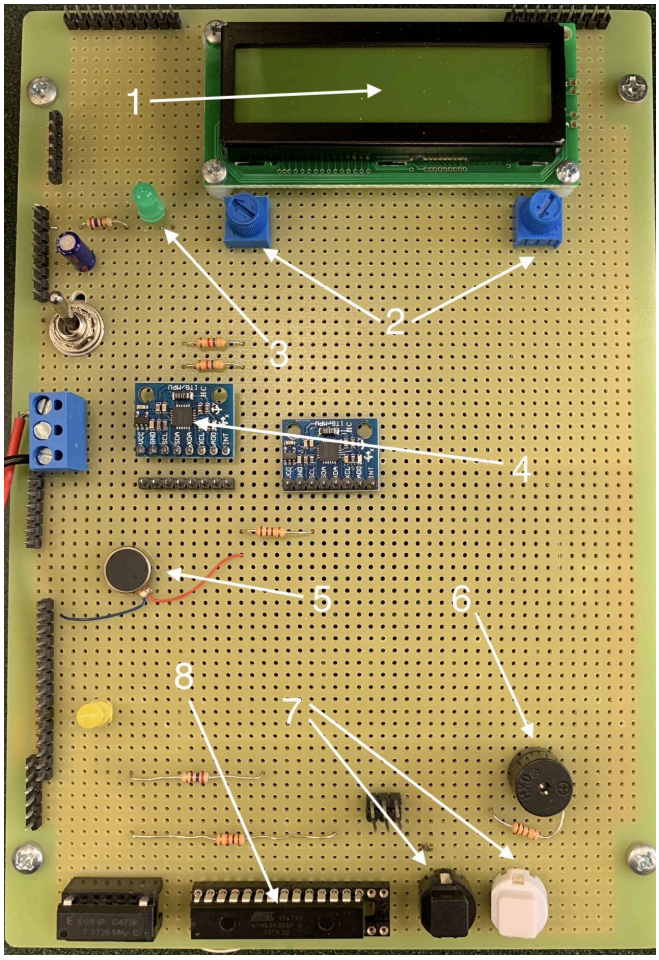
VI. Blockers Along The Way

We encountered two major technical challenges when developing the device. Though the solutions are simple, we hope they are useful for other developers.

- mpu6050 data reading issue: initially, we failed to read sensor data from mpu6050. After rounds of debugging, we figured out that the mpu6050 was installed on the wrong side of the board.
- LCD screen not lighting up: caused by bad soldering and lose connection between pins

VII. The Final Product

The figure below shows the circuit board of our final product. From the top to bottom, it features the following components:



1. LCD Display
2. Potentiometers for adjusting brightness and contrast
3. System status LED
4. Accelerometer and Gyroscope
5. Vibrator
6. Buzzer
7. 2x Buttons
8. Atmega328P Microcontroller

Changes from the original design:

The components we used for our fall detection device largely adhered to our original plan. We made some design changes regarding the placement of the components.

In our original design, we planned to use speakers to announce the detection of the fall, and subsequently, the alarm sound to draw passerbys' attention. A speaker is an intuitive acoustic device for output, so we chose it in our original plan. Nevertheless, after incorporating the

speaker into our system, we discovered that the sound of the speaker is not as loud as we anticipated, it will have difficulty penetrating clothing, especially in loud surroundings like campus. Since we want to allow the user to put the speaker in their pockets or bags, we needed some other method to notify the user and announce the fall.

This gives us an opportunity to rethink how we generate output. One idea is to create different levels of alarm to differentiate the system state using different components. Following this idea, we decided to use a vibrator and buzzer to provide a clear distinction between detection and alarm states. This makes it easy for users to identify the current state of the system. Also, the vibrator doesn't make any noise, which means our product will not create unnecessary noise right after a fall. Since even when a fall occurs, it's not always the case of a serious fall, a vibrator provides the user with a quiet period for assessing the situation.

Another change that we made was the LCD that we used. Originally we planned to use a slightly larger display, but its operating voltage is 3.3V, which is lower than the 5V power supply on our board. When testing the LCD, we failed to get a successful response. As a result, we uninstalled the board and mounted a smaller 5V LCD instead.

Product guidelines:

There are several guidelines that the project requirement highlighted. We will discuss how we adhere to those guidelines in our final project.

Firstly, the guideline specifies that our product must be made for consumer usage and a clear reason for being. This means the product should be intended for everyday use instead of for business purposes and solving real-life problems. For riders of bikes, scooters, or skateboards, our detector could be placed in their backpack when they come to campus. If they were to fall on campus and lose consciousness, the buzzer would draw attention from passersby. Elderlies are usually susceptible to falls since their mobility is impaired by old age. If they carry our fall detector along with their belongings, they will be able to draw attention even if they can't find their handset. Children who are playing in the playground could carry the fall detector on their back when they engage in intense activities. If they fall, assuming that they can't dismiss the fall

alarm due to their lack of literacy, the buzzer noise will call for guardians' attention and come to the child's rescue.

Innovativeness and ease of use are also included in the guidelines. After comprehensive research, we found that most fall detection tools are integrated as part of wearable devices like smartwatches and smartphones. However, many elderly people and young children can't use those multipurpose devices. This means users who would benefit from our product the most do not have access to a similar product yet. Our product is innovative because it provides an easy-to-access product that strips away other complex functionalities. The ease-of-use aspect of this product is fulfilled by the simple user interface that we implemented. To control the device, the user only needs to control two buttons to input information into the device. The LCD also presents clear and concise instructions to guide and inform the user with necessary information. This makes our product extremely simple to operate. The simplicity of operation also contributes to our product's ergonomics, which is another part of the guideline. Since the buttons are color-coded with black and white, it's easy for users to follow the displayed instructions and find the right button to press. The design of our product ensures accessibility for all customers.

One of the fundamental traits of our design is its real-time response. When a fall is detected, the vibrator immediately starts operating, notifying the user that the acoustic alarm will start imminently.

Product requirements:

In addition to the guidelines, there are also many requirements outlined in the original project briefing as well. The first of which pertains to the cost-effectiveness of the product. The most expensive component that we installed on our board only costs 12.95 dollars, with a total cost of 27.9 dollars. This means our product's price is very competitive for what it can accomplish.

Another requirement is that our product should have at least five inputs and/or outputs of different types. In our design, the three inputs are the 2 buttons, the gyroscope and accelerometer,

and the potentiometer. On the other hand, the three outputs are a vibrator, buzzer, and LCD. Therefore, in total, there are six input/output components in our system.

The last relevant requirement for our project is immunity to power outages. In the context of our product, the user's information should be retained even after powering off. To enable this functionality, we used the EEPROM of our ATMEGA328P microcontroller. Upon startup, the user has the choice of resuming the detection with a saved profile from the EEPROM.

VIII. Engineering Standards

In our product, we used single-wire communication to transfer information to and from simple components: buttons, vibrator, and buzzer. However, for more heavy-weight components, namely LCD and accelerometer & gyroscope, we had to use some more complicated communication protocols.

Inter-Integrated Circuit (IIC): Since the accelerometer & gyroscope module that we chose, MPU 6050, has only two pins for communication, it relies on IIC, a serial communication protocol, to transfer data to and from the microcontroller. On runtime, it provides acceleration and orientation information to the microcontroller periodically. The microcontroller then uses that information to determine if a fall has occurred.

Parallel Communication: The LCD has 8 pins for data transmission. This allows the LCD to display the information with minimal delays. For our microcontroller, we dedicated PINB0 to PINB7 to interface with the LCD. On runtime, the microcontroller sends a byte of data simultaneously to the LCD, which then decodes and displays the data.

IX. Future Improvements

The current model is only a prototype that is subject to future improvements. There are three main areas that we will focus on for our next iteration: Fall detection algorithm, size, and reliability.

Fall detection: Our fall detection method at the moment only considers vertical (z-axis) acceleration as the defining characteristic of a fall event. However, this means for users who have very dramatic movements, such as children, bikers, and skaters, false alarms might be a concern. Nevertheless, it's extremely difficult to come up with a clear definition of what a fall means in terms of acceleration and orientation data in 3D space because almost every fall has its unique characteristics.

One potential solution is to use machine learning, more specifically, reinforcement learning. Since we already have a method to detect falls, we can improve it by providing labeled training data. The training data can be provided in the form of repeated experiments where we simulate various falling and normal scenarios. This will inform the machine learning model on how to distinguish between a sudden movement and an actual fall.

Size: currently our product is much larger than the size of an adult hand. Furthermore, it features a lot of hard and sharp protrusions that might potentially become hazardous for users who are vulnerable to collisions. These considerations will be brought into our next design, which we anticipate to be much smaller. Specifically, the next design should have a palm-size round encasing that encloses the circuit board within it.

Reliability: The design at this stage has no water-resistance and collision-resistance capability. However, our product could potentially be used in places that have rugged terrains. For example, a child could take our product with them as they play in a park full of puddles and mud. Our product, without a hard and water-proof casing, is vulnerable in such conditions.

X. Appendices

References

[1] K. Smith, “The importance of fall detection apps for seniors,” Seniors Guide, https://www.seniorsguide.com/technology/the-importance-of-fall-detection-apps-for-seniors/

[2] R. Tanwar, N. Nandal, M. Zamani, and A. A. Manaf, “Pathway of trends and technologies in fall detection: A systematic review,” MDPI, https://www.mdpi.com/2227-9032/10/1/172

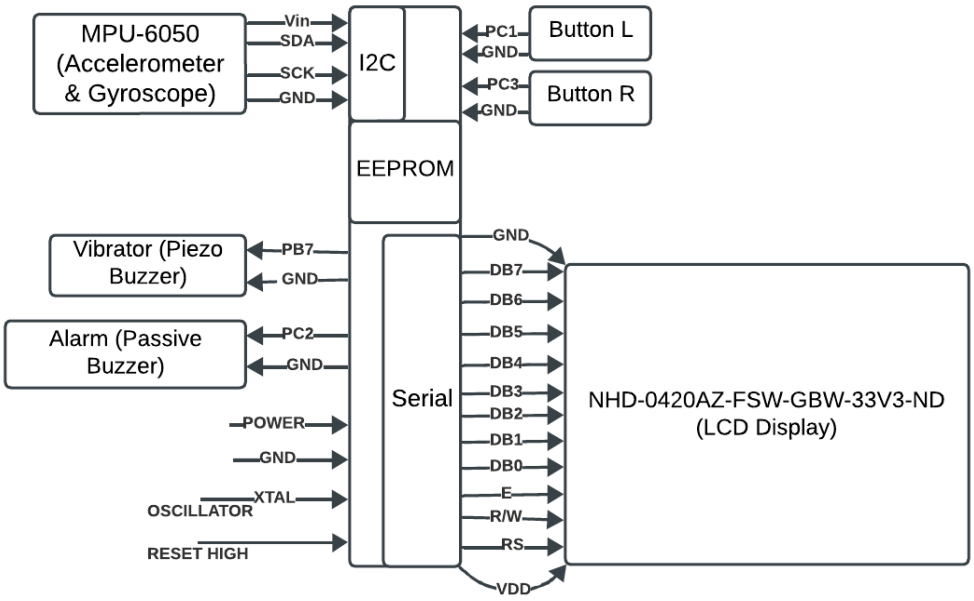
[3] J. Rowley, “The Importance of Fall Prevention Technology,” RF Technologies, https://www.rft.com/the-importance-of-fall-prevention-technology/

Parts list and cost details

Part	Quantity	Unit Price	Component Cost
Button	2	$0.35	$0.7
Accelerometer & Gyroscope	1	$12.95	$12.95
Buzzer	1	$1.75	$1.75
Vibrator	1	$1.95	$1.95
LCD	1	$4.95	$4.95
Microcontroller	1	$2.80	$2.80
LED	1	$0.3	$0.3
Potentiometer	2	$1.25	$2.5
		Total Cost	$27.9

Figures

Schematics:



State Machine: