Project work - August 15

BlackBox For Motorcycles

Title :

BlackBox for Motorcycles, a device that can be used in motorcycles which detects crashes and stores logs of a motorcycle.

Team Members:

- Sanjith S J - *CB.SC.U4CSE24544*
- Lohit G - *CB.SC.U4CSE24522*
- Madan M - *CB.SC.U4CSE24523*
- Nitin Pranav R - *CB.SC.U4CSE24539*

Summary:

This project is about creating a **black box** for motorcycles that records important ride data, such as acceleration, tilt, and movement patterns and saves the last few seconds before and after a crash.

We use an **L3G4200D gyroscope and ADXL335 3-axis accelerometer** to detect motion and sudden changes in acceleration. An **Arduino Uno** collects this data in real time. If the system detects a crash, it instantly saves the recent ride data to an **SD card module** for later analysis and using a **GSM MODULE**, it also calls an emergency contact in case of an accident.

The main idea is to provide an affordable prototype for motorcycle safety monitoring, which can help in accident investigations or performance analysis. This is a **breadboard-based prototype**, powered directly from USB, so it's easy to test and modify without extra power systems or enclosures.

Problem Statement:

Most motorcycles don't have any built-in way to record what happened before an accident. This means crucial information like speed, tilt, and movement just before the crash is lost.

High-end vehicles have event data recorders, but these are expensive and complex. For motorcycles, there is a need for a **low-cost, easy-to-build crash data recorder** that can store the last few seconds of data when a crash occurs.

Our project aims to fill this gap by making a **motorcycle black box prototype** using Arduino, affordable sensors, and open-source code.

Requirements:

Hardware Requirements:

- Arduino UNO
- L3G4200D gyroscope
- ADXL335 3-axis accelerometer
- SD Card Module
- MicroSD Card
- Breadboard
- Jumper Wires
- USB Cables

Software Requirements:

- Arduino IDE
- Arduino L3G4200D and SD libraries
- Serial Monitor

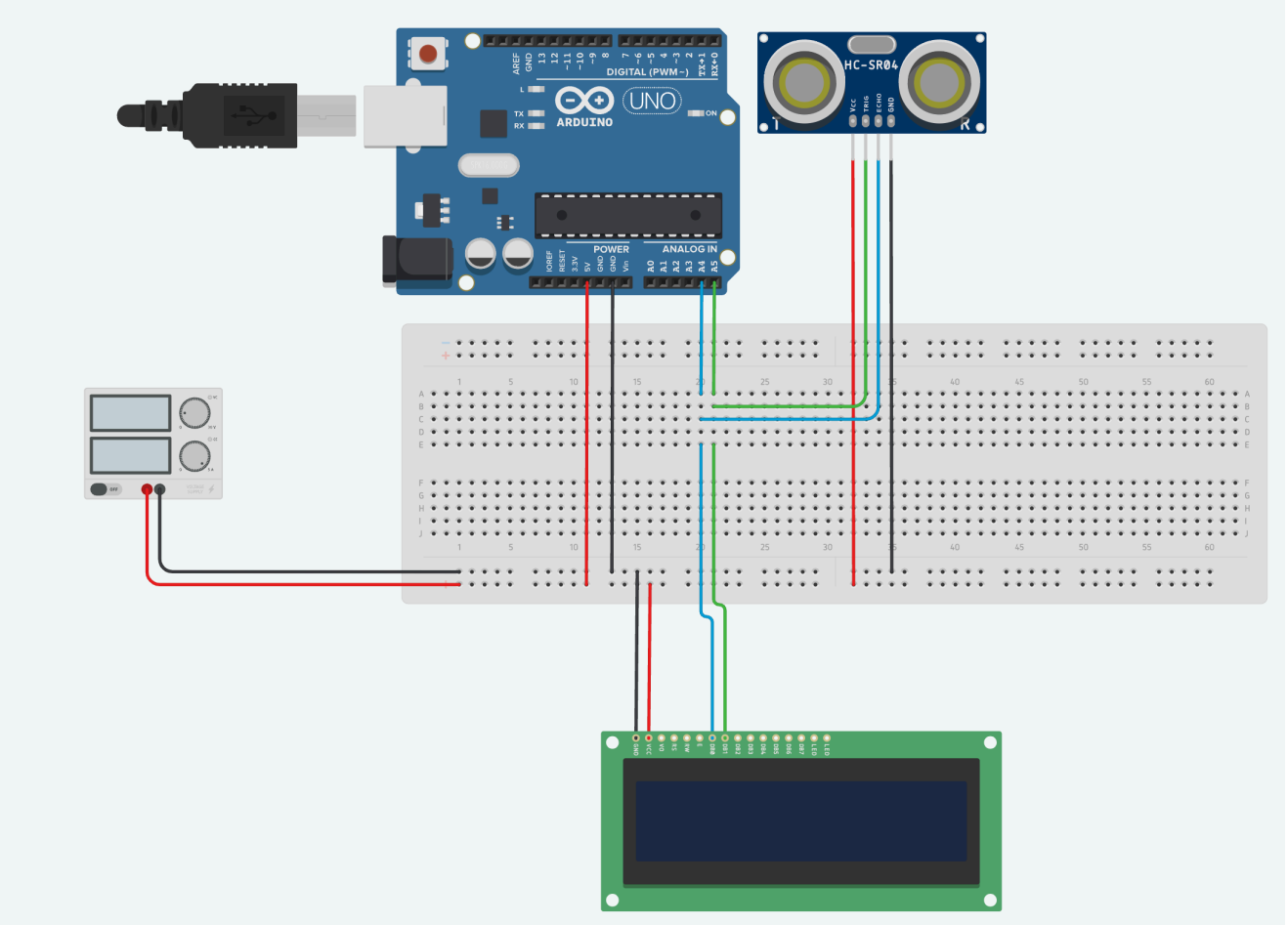
How is it all connected together :

The **L3G4200D and the ADXL 335 sensor** is the “eye” and “inner ear” of the system it senses acceleration and rotation in real time. It connects to the Arduino Uno through I²C communication (SDA and SCL pins).

The **Arduino Uno** is the “brain” of the system. It constantly reads motion data from the MPU6050 and stores it in a **circular buffer** in memory, which always contains the last few seconds of movement history.

The **SD Card Module** acts as permanent memory. When a crash is detected, the Arduino writes the buffer’s data into a CSV file on the SD card.

The system is powered via USB during this prototype phase, making it easy to test without worrying about batteries or bike integration.



The Step-by-Step Logic (Flow):

1. **Start:** The Arduino powers on.
2. **Get Ready:** The setup() function initializes the MPU6050, SD card, and sets up variables.
3. **Read Motion Data:** The Arduino continuously reads acceleration and gyroscope values from the L3G4200D and ADXL335 modules.
4. **Store in Circular Buffer:** Each new reading overwrites the oldest one in memory, keeping only the most recent few seconds of data.
5. **Check for Crash:**
   - If acceleration magnitude is above a set threshold (e.g., > 2.5g) or there's a sudden tilt change, it's flagged as a crash.
6. **Save Data:** If a crash is detected, the contents of the circular buffer are written to the SD card as a CSV file.
7. **Continue Monitoring:** The loop repeats forever, ready to capture the next event.

C Concepts We Used:

- **Structures:** Used to group motion readings (time, acceleration, gyroscope) into one unit.
- **Arrays:** Implemented as a circular buffer to store a fixed number of recent readings.
- **if-else Statements:** Used to detect whether a crash condition has occurred.

- **Functions:** Separate functions for reading the sensor, detecting crashes, and writing to SD card.
- **Input/Output Commands:** Wire.read() and Wire.write() for I²C communication, SD.open() for file operations.
- **Math Functions:** Calculating acceleration magnitude using sqrt() for crash detection.

What We Expect to See:

- Under normal riding (or simulation by shaking the sensor lightly), the system keeps logging data to memory but does not save to the SD card.
- When a sudden jolt or tilt is simulated (e.g., dropping the sensor), the system immediately writes the last few seconds of motion data to the SD card in a CSV file.
- Opening the CSV file on a computer shows a timeline of acceleration and rotation just before the crash.

Additional Features:

- A python flask application will monitor the acceleration and rotation values in real time and will display a dashboard that helps for debugging and logical improvements in the development stages.

Conclusion:

This project demonstrates how low-cost electronics and simple C programming can create a functional prototype of a motorcycle black box.

Even though this version runs only on a breadboard and USB power, it lays the groundwork for a real-world motorcycle accident data recorder.

In the future, we can add a GPS module to log speed and location, and a GSM module to send accident alerts. We could also design a permanent PCB version for actual motorcycle installation.

Sources and Further Reading:

1. Arduino Official Website. (2025). *Arduino Language Reference*. https://www.arduino.cc/reference/en/
2. InvenSense. (2023). *MPU6050 Product Specification*.
3. Autodesk Tinkercad. (2025). *Official Website*. https://www.tinkercad.com