# Implementation

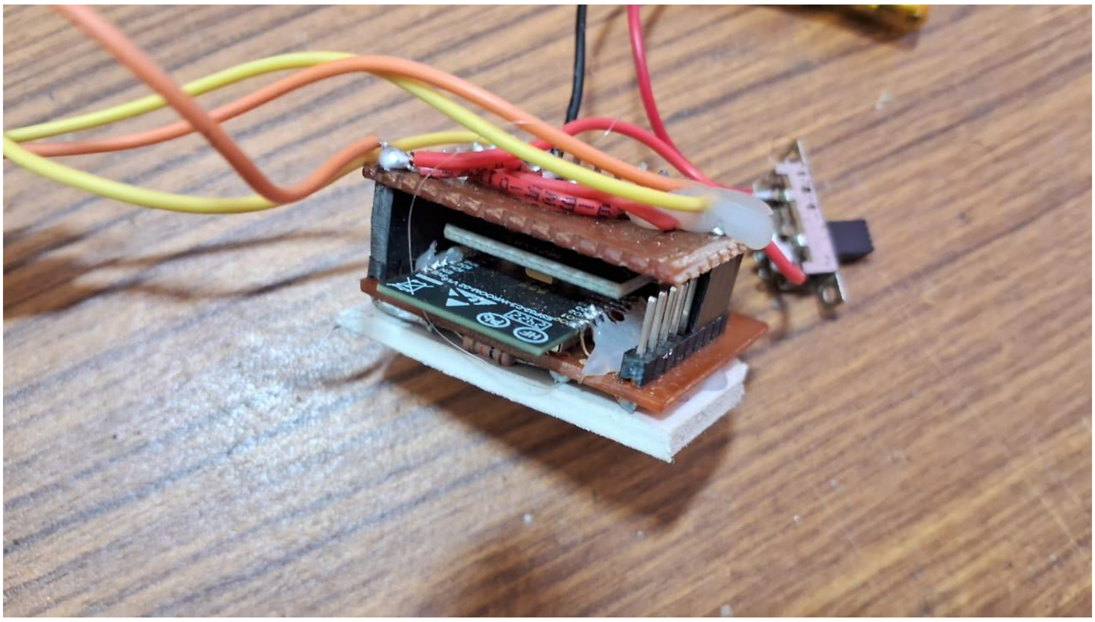
**Name:** Mayank Baldania
**Enroll :** 92310133011
**Branch :** BTech ICT

# Table Contents

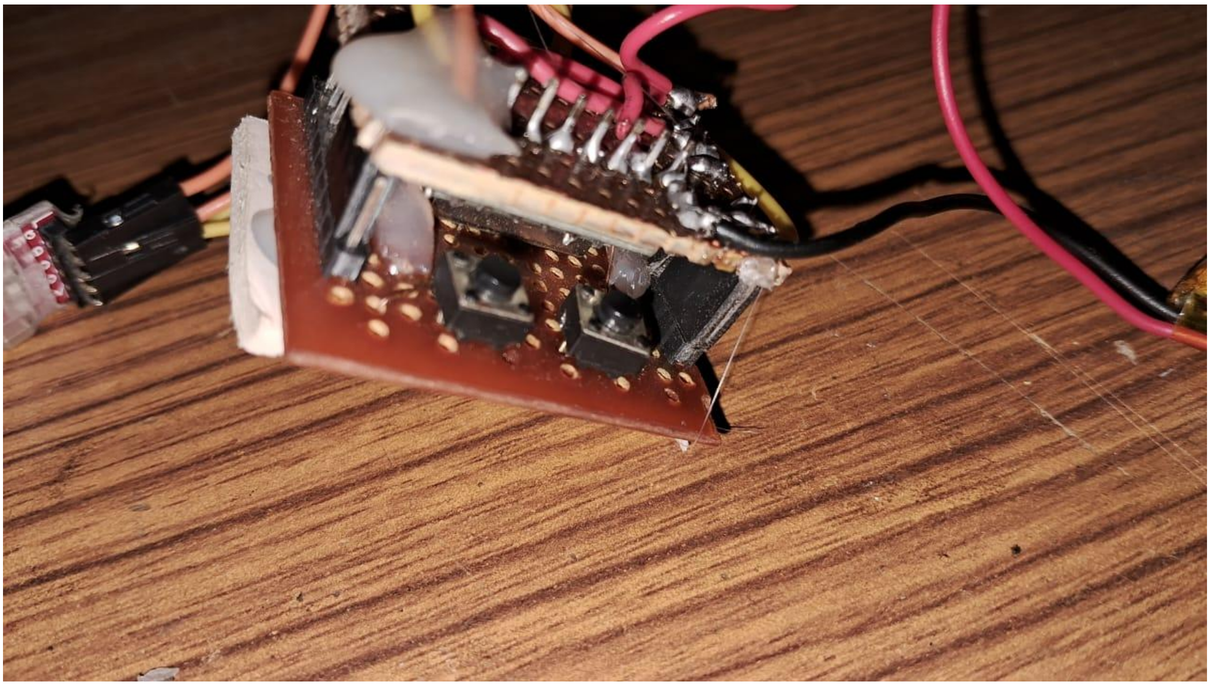
# 1. Project Objectives

- Design and develop a low-cost prototype using ESP32/ESP8266 and sensors to demonstrate Newton's Laws of Motion in real-time.
- Enable real-time data collection and visualization of motion parameters (e.g., acceleration, velocity, and force).
- Provide hands-on learning experiences for students, making physics concepts easier to understand compared to only theoretical teaching.
- Integrate IoT and web technologies (WiFi, WebSocket, Node.js, HTML, and Chart.js) to build an interactive system.
- Support teachers with easy-to-use tools that allow classroom demonstrations without expensive laboratory equipment.
- Improve learning engagement by making experiments interactive, visual, and accessible through any web browser.

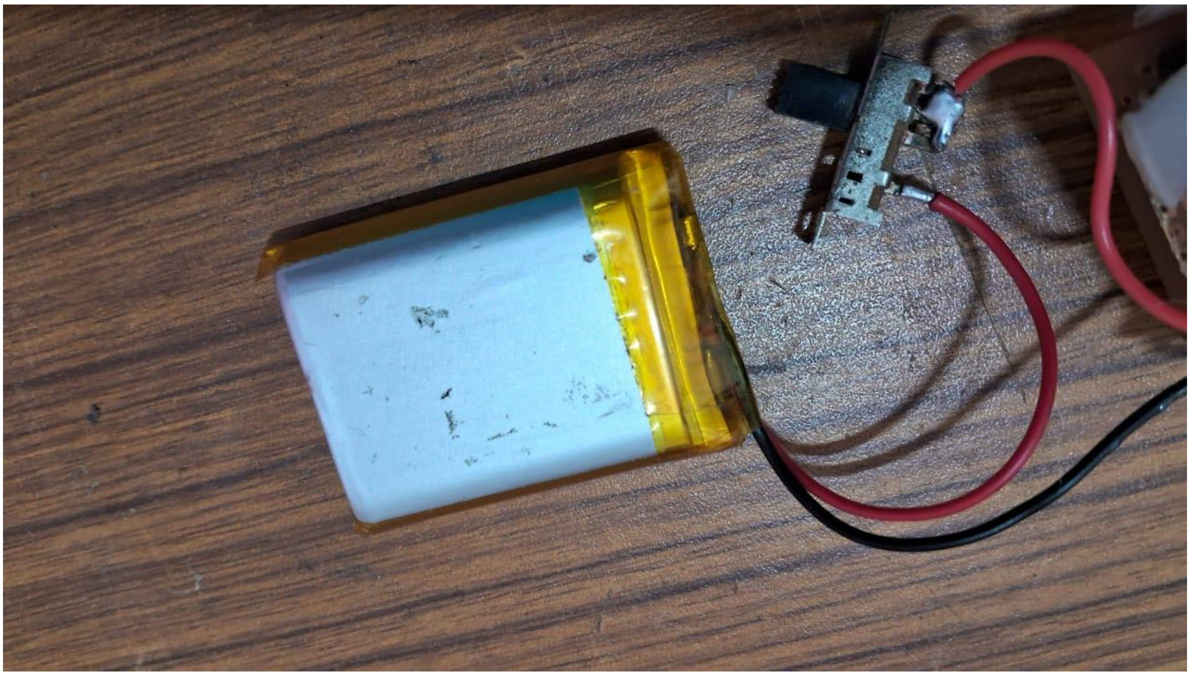
# 2. Hardware Implementation



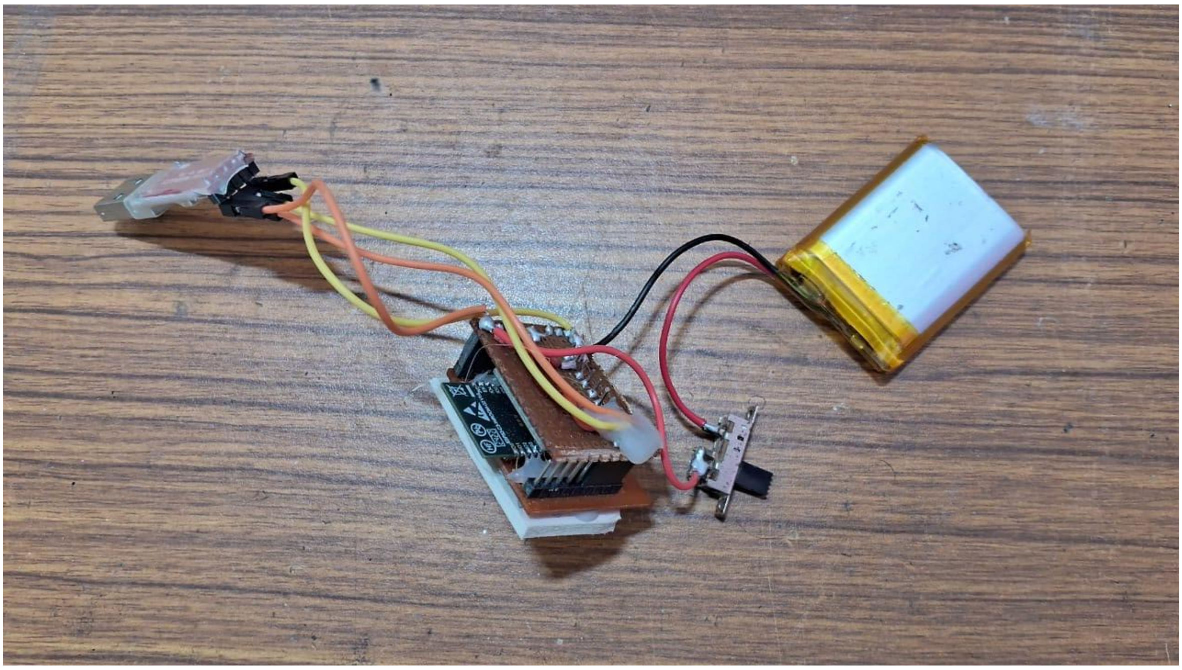
Using ESP32C3WROOM02 & ADXL345 Accelerometer, we developed a very compact size of circuit module from scratch without using any development board like NodeMCU etc..



Implemented 2 switch for Program FLASH configuration in ESP32C3WROOM02.



3.7V LiPo Battery & Switch

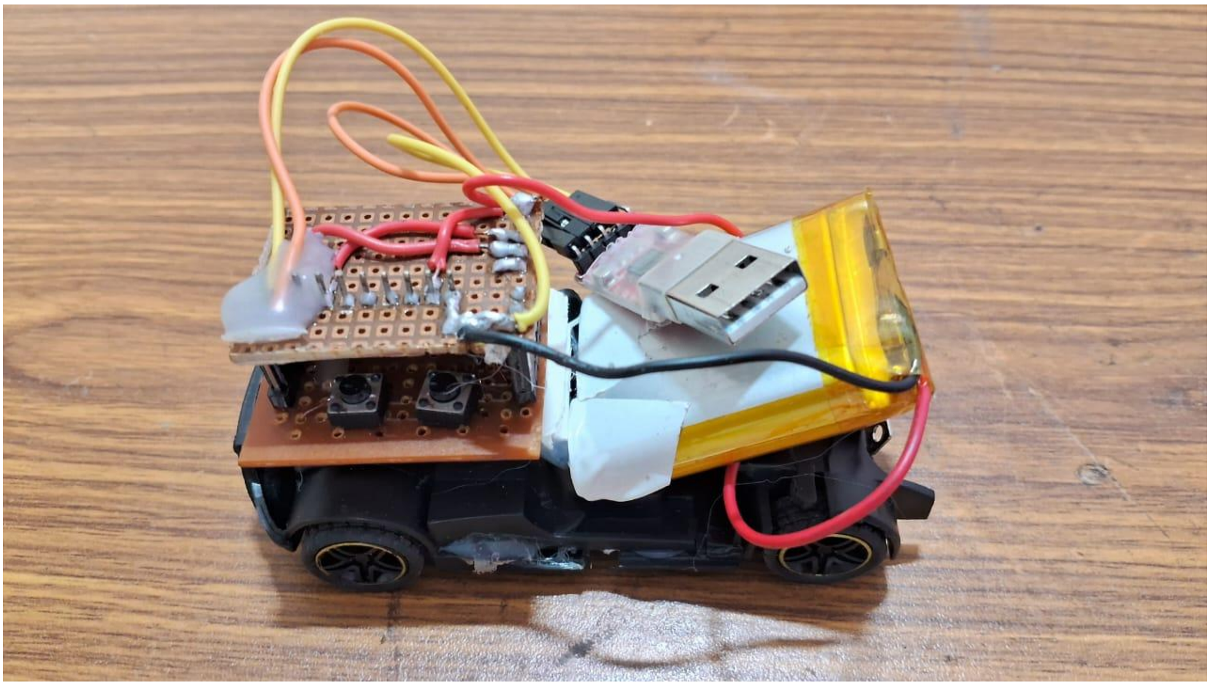


Overall Circuit with excluding USB to TTL for Flash Program.

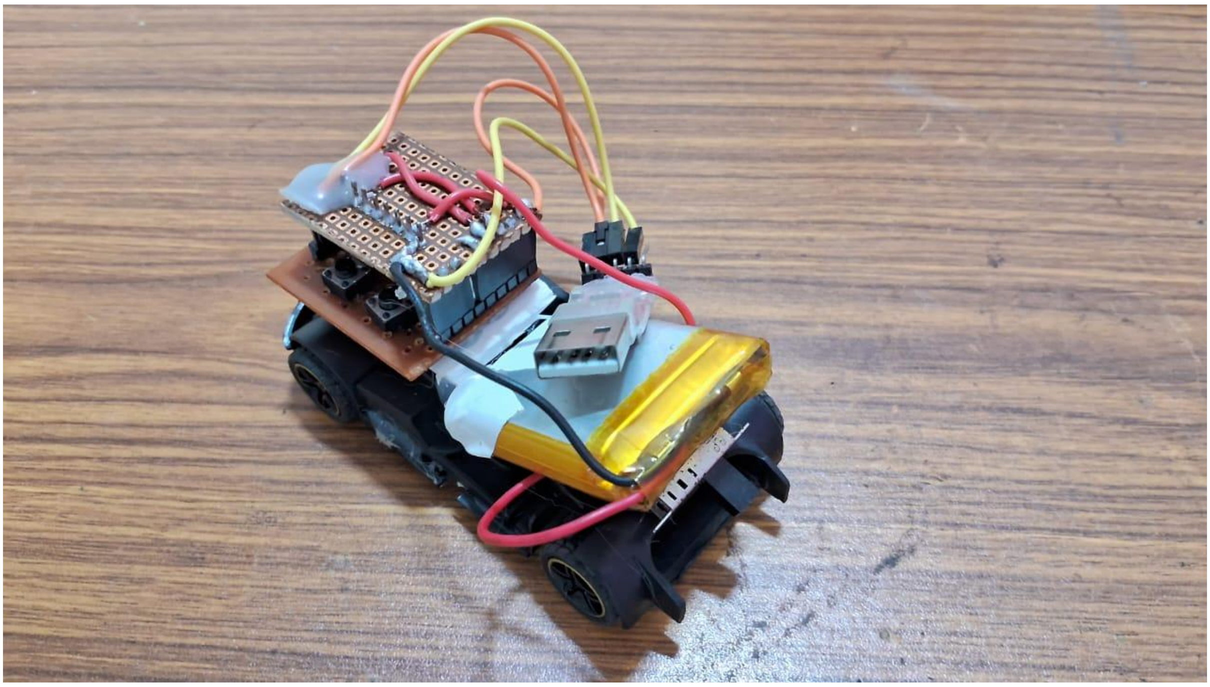
Simple Small Toys Car.

Remove upper body part of Toy Car.

Grinded & Cut the upper part for fitting Circuit Module.



Mount the Circuit module in this toy car lower body.



Final project of Real time Acceleration Newtonia Project.

# 3. Firmware Implementation

Code Source (Github) : https://github.com/MayankBaldania/Newtonia-Project/blob/2729bc8352da9520848a1cacf3cf3b27a72bb21f/Final_Code.ino

# 4. Functionality

## 4.1 Key Technology

- ESP32-C3: A microcontroller with integrated WiFi, used to read accelerometer data, & host a web server.
- WiFi.h Library: Manages WiFi connectivity for the ESP32-C3.
- WebServer Library: Runs an HTTP server on port 80 to serve a dynamic HTML webpage with a real-time graph.
- WebSocketsServer Library: Operates a WebSocket server on port 81 for low-latency, real-time data streaming to the client.
- ESPmDNS Library: Enables mDNS (multicast DNS) to access the ESP32-C3 via a local hostname (http://newtonia.local).
- Wire.h Library: Facilitates I2C communication with the ADXL345 accelerometer to read acceleration data.
- ADXL345 Accelerometer: A 3-axis accelerometer used to measure acceleration along the X-axis.
- Chart.js: A JavaScript library on the client side to render a real-time line graph of acceleration data.
- HTML/CSS/JavaScript: Provides a responsive webpage with a graph, control buttons, and a force calculation section based on Newton's second law.

## 4.2 Key Algorithms and Logic

- ADXL345 Data Acquisition:
  - The function reads 6 bytes from the ADXL345's register (X, Y, Z axes, though only X is used).
  - Converts raw X-axis data (16-bit, two's complement) to acceleration in $m/s^2$ using the formula: (x / 256.0) * 9.81, where 256 LSB/g is the sensitivity for ±2g range, and 9.81 $m/s^2$ is the gravitational constant.

- Moving Average Filter:
  - A circular buffer (movingAvgBuffer) of size 2 stores recent X-axis readings.
  - The movingAvgSum tracks the sum of buffer values, updated by subtracting the oldest value and adding the new one.
  - The average is computed (movingAvgSum / MOVING_AVG_SIZE) once the buffer is full, smoothing out short-term noise.
- Exponential Moving Average (EMA) Filter:
  - Applies EMA to the moving average output: X_filtered = (EMA_ALPHA * X_out) + ((1.0 - EMA_ALPHA) * X_filtered), where EMA_ALPHA = 0.1 controls smoothing (lower alpha = smoother output).
  - This further reduces noise for a stable graph display.
- Real-Time Graphing:
  - The client-side JavaScript parses JSON WebSocket messages ({"x": value}) and updates a Chart.js line graph.
  - Maintains a sliding window of 300 data points, removing the oldest point when the limit is reached to optimize performance.
- Force Calculation:
  - On the client side, Newton's second law (F = m * a) is applied with a fixed mass of 0.07 kg (70g).
  - The acceleration input (accelInput) allows manual entry, and the force is computed and displayed (force = massKg * accel).