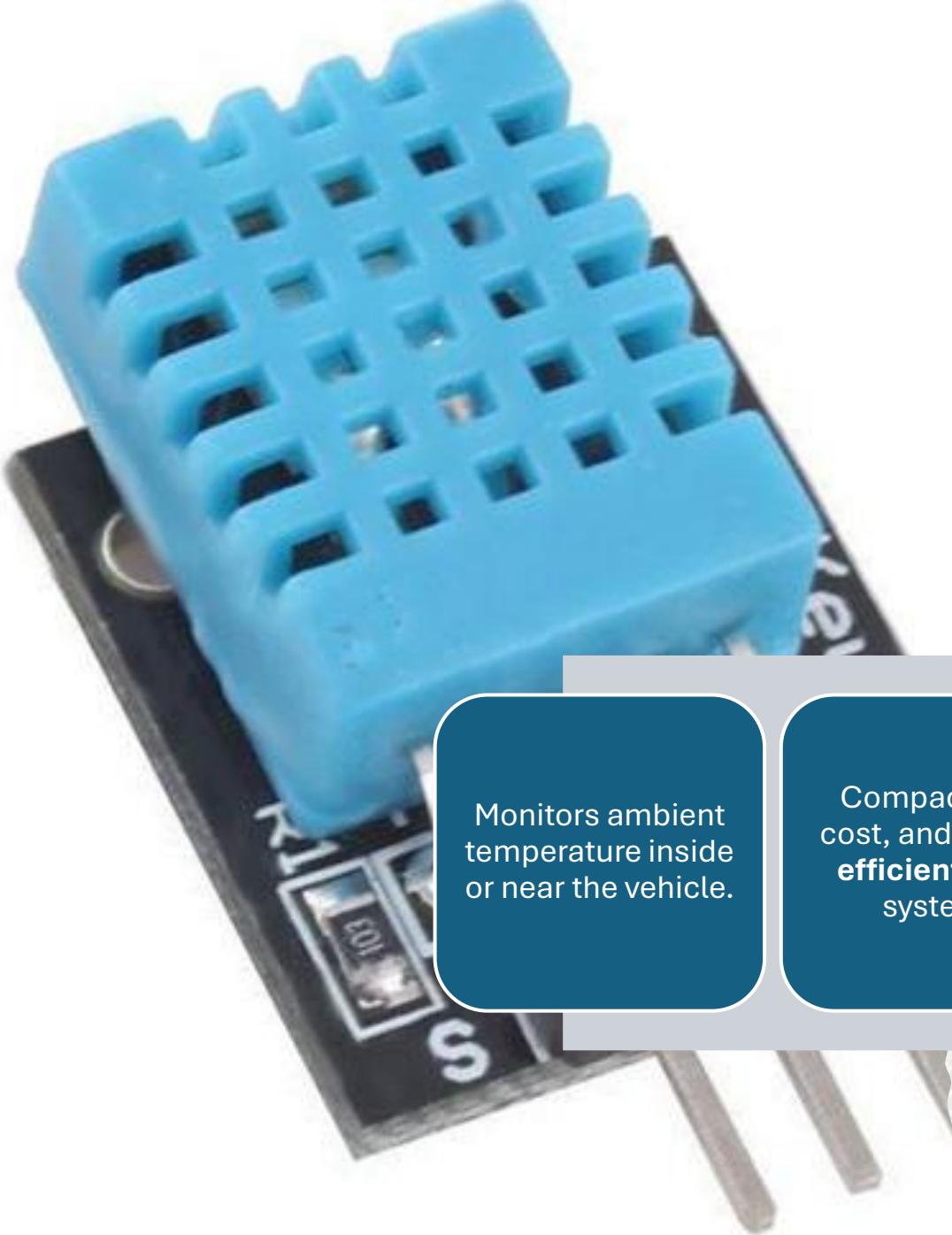


- ✓ Sensors
- ✓ Hardware components
- ✓ Software & Programming Tools
- ✓ Cloud & Web Dashboard Platform
 - ✓ Libraries & API
- ✓ Development Environmentnt
- ✓ Security Tools & Methods

Equipment List





Sensors: DHT11 Temperature

Monitors ambient temperature inside or near the vehicle.

Compact, low-cost, and **energy-efficient** for IoT systems.

Can provide early warning if the system detects high temperature or overheating (preventive safety).

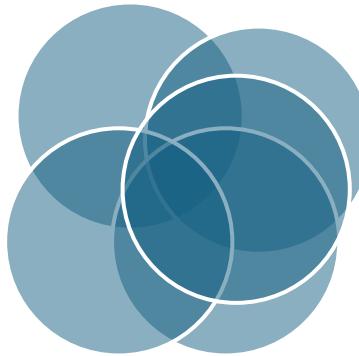
Stores temperature data in the cloud (Firebase) for future analysis or insurance reports.

Used to show temperature readings on the OLED display and dashboard.



GPS Module NEO-6M

Used to send **real-time coordinates** to Firebase and include in the Twilio SMS alert.



Essential for locating the accident site quickly and helping emergency responders.

Provides accurate **latitude and longitude** location of the vehicle.

Allows the system to **locate the exact accident position**.

Helps the **cloud backend (Google Geocoding API)** convert coordinates into a readable street address.



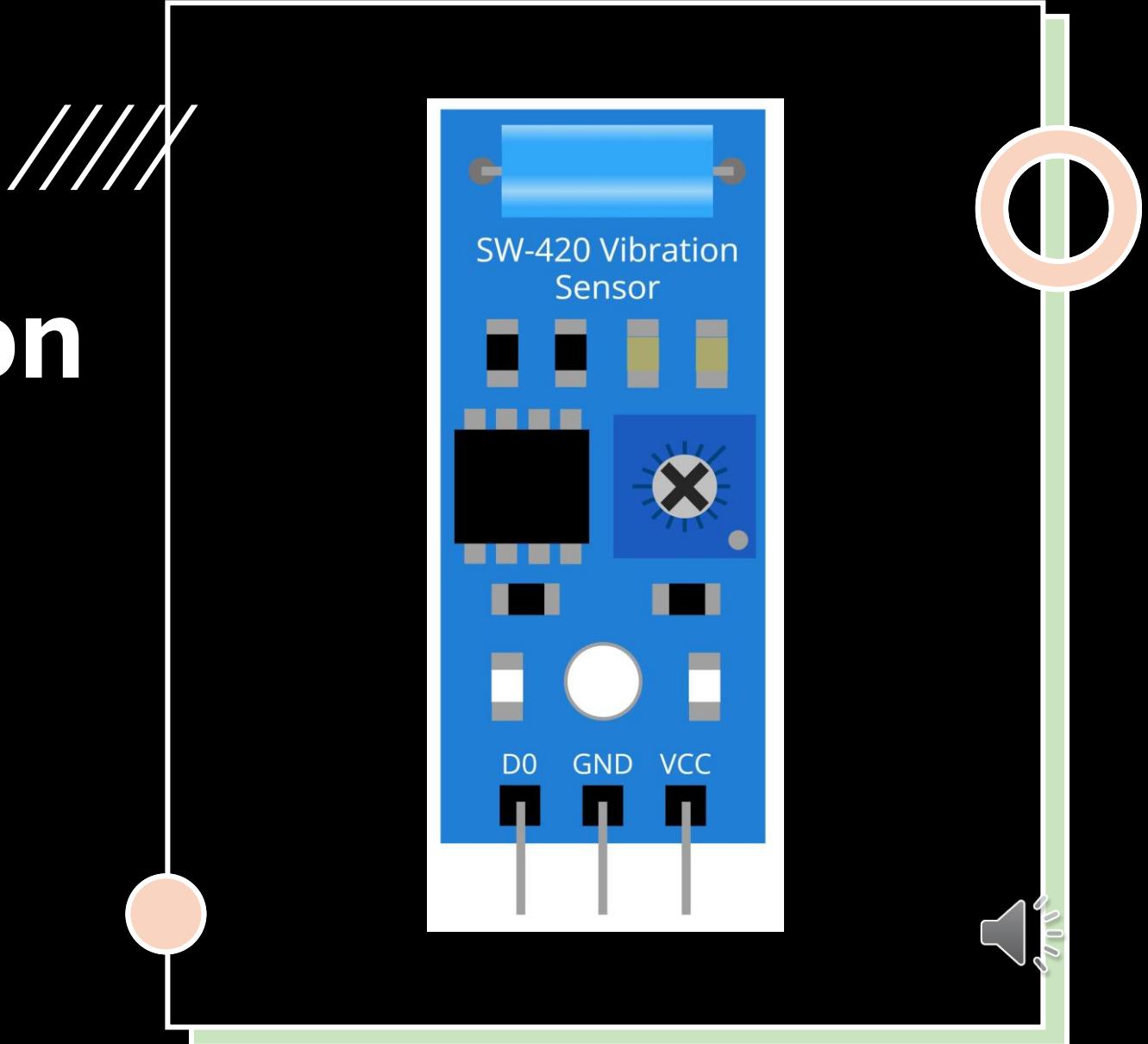
SW-420 Vibration Sensor

Detects mechanical vibrations or shocks when a collision occurs.

Works together with the MPU6050 to confirm an accident through multi-sensor validation.

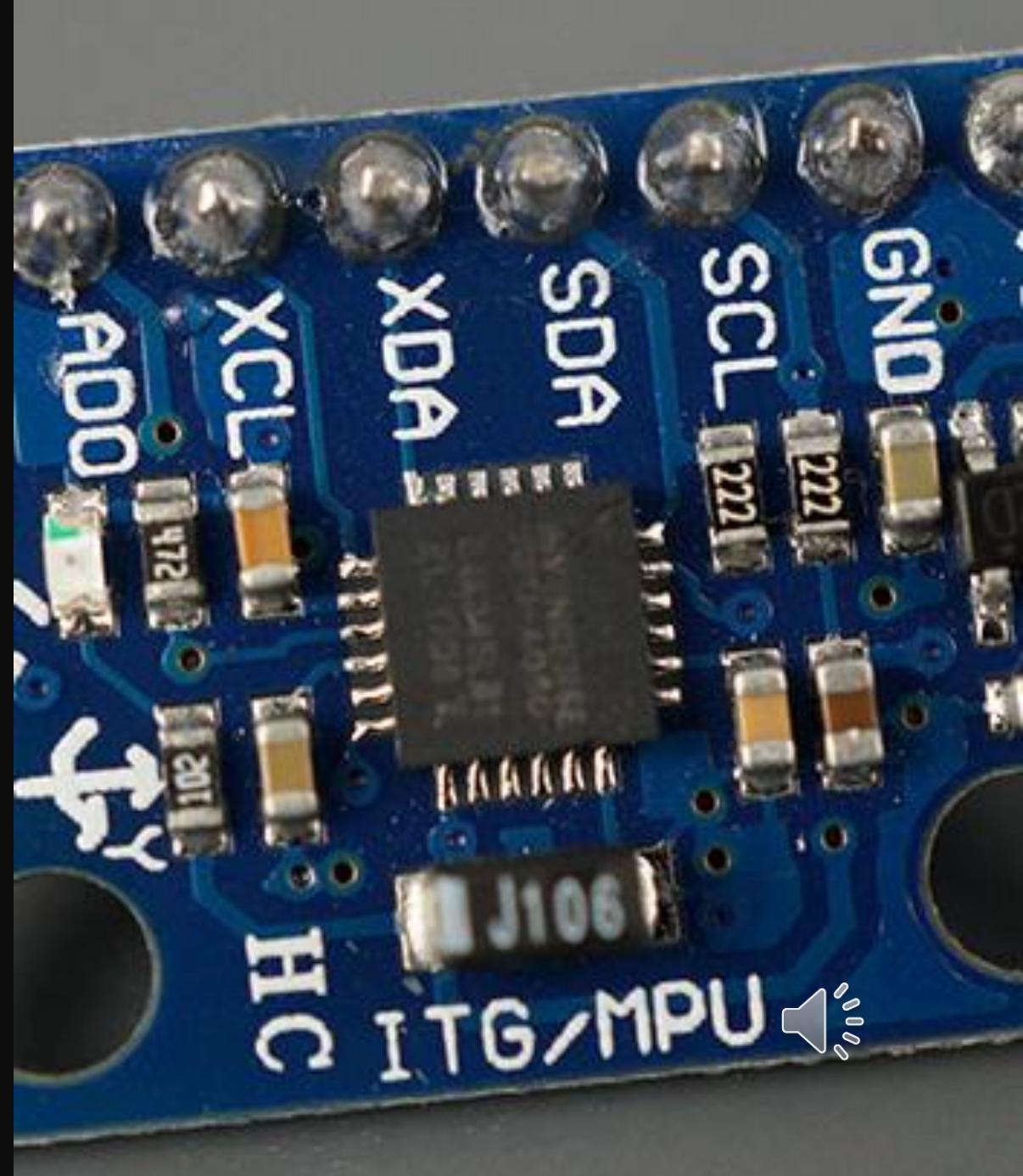
Provides a quick and simple signal output when vibration exceeds a set threshold.

Increases the reliability of accident detection by filtering out false triggers.



MPU6050 Accelerometer + Gyroscope sensor

- Detects sudden impact, tilt, or high acceleration — used to identify when an accident has occurred.
 - Measures changes in motion (acceleration) and orientation (gyro) to detect abnormal vehicle behavior.
 - Helps differentiate between normal vibration and serious collisions.
 - Provides accurate movement data that triggers the **accident detection** event in the system.
 - Provides **real-time motion data** to determine accident severity.
-



Hardware Components

Arduino MKR WiFi 1010

Built-in **Wi-Fi module (NINA-W102)** allows direct cloud connectivity without external shields.

Powered by **ARM Cortex-M0+ 32-bit microcontroller**, providing better performance than Uno or Nano.

Fully compatible with **Arduino IoT Cloud** and **Google Cloud HTTPS communication**.

Provides **secure encrypted communication (TLS 1.2)** needed for sending sensitive accident data.

Multiple digital/analog pins support sensors like **MPU6050, DHT11, GPS, OLED, and buttons**.

Integrated **USB and battery charging circuit** — easy to power from USB or Li-ion battery.

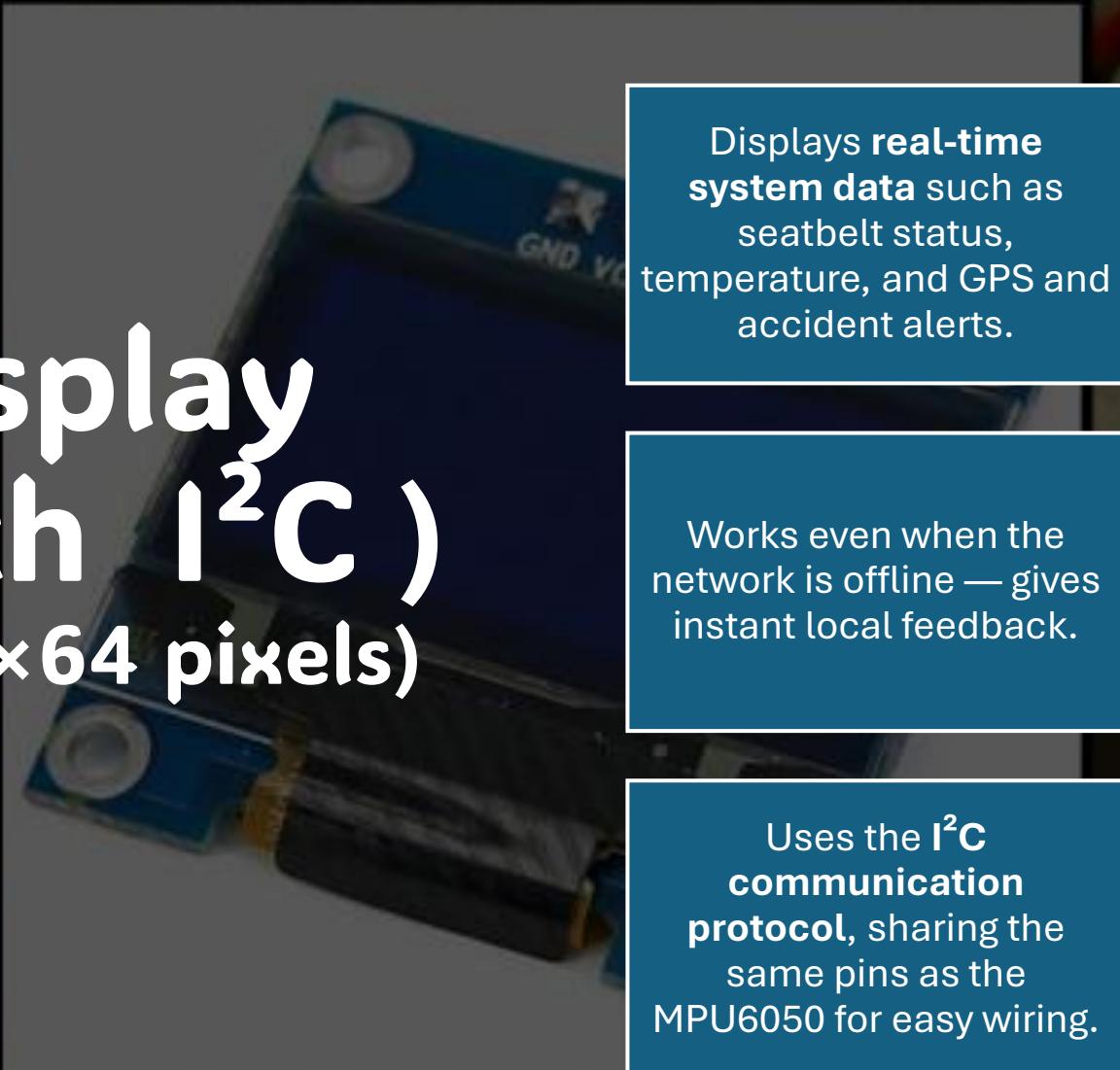
Simplifies the hardware design — all network, logic, and power systems are built-in.

Excellent support through **Arduino IDE** and libraries for WiFi, JSON, and HTTPS.

Stable, reliable, and designed for **IoT + cloud-based applications** like this accident alert system.



OLED Display (0.96 inch I²C) (SSD1306, 128×64 pixels)



Displays **real-time system data** such as seatbelt status, temperature, and GPS and accident alerts.

Shows **status messages** like “Accident Detected,” “Alert Canceled,” or “Wi-Fi Connected.”

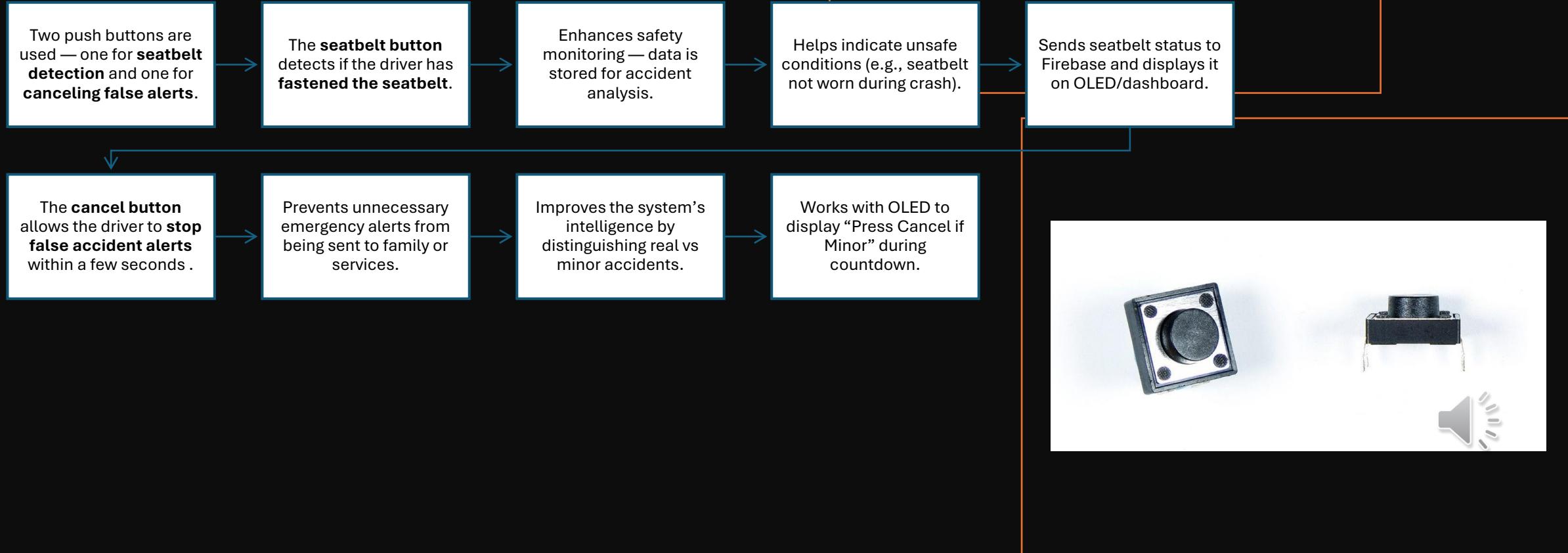
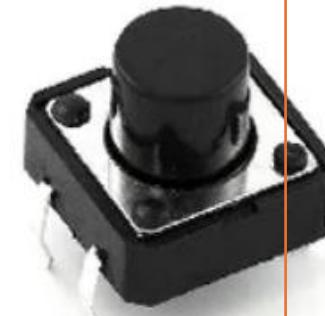
Works even when the network is offline — gives instant local feedback.

Helps the driver easily understand the system status (Normal / Alert / Sending).

Uses the **I²C communication protocol**, sharing the same pins as the MPU6050 for easy wiring.

Compact (0.96") and **low-power**, ideal for embedded and vehicle projects. 

Push Buttons (Seatbelt Detection & Cancel Button)



Buzzer (Alert Notification)

Provides an **audible alert** when an accident is detected.

Warns the driver during the **25-second cancel window** (false alarm prevention).

Provides immediate feedback even if the internet is unavailable.

Used with the OLED to ensure both **sound + visual** warnings.

Beeps when the system is about to **send the emergency SMS** via Twilio.

Gives **sound feedback** for events like “*Alert Sent*” or “*Canceled Successfully*.”



Piezo Buzzer

Supporting Hardware

Breadboard – For prototyping and connecting all modules.

Jumper Wires (Male–Male, Male–Female) – For electrical connection between Arduino, sensors and breadboard.

Resistors – pull-down for buttons and for LEDs.

USB Cable – Used for programming and power supply



Software & Programming Tools

Arduino IDE – Main development environment to write, compile, and upload code to the MKR WiFi 1010.

Arduino Libraries:

- WiFiNINA,
- ArduinoHttpClient,
- ArduinoJson,
- DHT,
- Adafruit_MPU6050,
- TinyGPS++,
- Adafruit_SSD1306 – Enable Wi-Fi, HTTP, sensor, and OLED functions.

Python 3.10 + Flask Framework– Backend web server to handle HTTPS requests, process data, call Twilio API, and update Firebase.



Cloud & Web Dashboard Platforms

Google Cloud Run (FastAPI Backend): ---Handles HTTPS requests from the Arduino board.

Firebase Hosting (HTML/CSS/JS): Hosts and displays the web dashboard (frontend interface) for remote users.

Firebase Firestore: Cloud database for storing accident incidents, alerts, and history.

Twilio API: Sends SMS alerts automatically to family members or the owner after accident confirmation.



Development Environment

Windows 10/11 or Linux	System for coding and running backend.
Google Chrome / Edge	Testing web dashboard interface.

