

★ 1. WiFi + Device Setup

- ESP32 connects to a WiFi network.
 - Loads device ID, firmware version, memory info, etc.
 - Prepares the device for cloud communication.
-

★ 2. CAN Bus Initialization

- The code configures the ESP32's CAN controller (TWAI).
 - Sets CAN speed (like 250k/500k).
 - Starts receiving CAN frames continuously.
-

★ 3. Reading CAN Data

The ESP32:

- Reads each incoming CAN frame (ID + data bytes).
 - Converts it into a readable JSON structure.
 - Extracts important parameters (RPM, temperature, pressure, hours, etc.) depending on your machine.
-

★ 4. Logging to SD Card (Local Storage)

- Creates a daily folder (like 2025-11-13/).
 - Saves each CAN frame to a .log or .csv file.
 - Ensures no data is lost even if Internet is down.
-

★ 5. Live Data Upload to Cloud (Flask Server)

- Sends selected parameters to the Flask API:
`/api/live-data`
 - This updates the dashboard in real time using WebSockets.
 - Includes:
 - `device_id`
 - `parameters`
 - `GPS if available`
 - `status`
-

★ 6. Memory Monitoring

- Code checks free/used SD card space.
 - Sends it to the server so dashboard can display storage info.
-

★ 7. Reconnection + Error Handling

- Auto reconnects to WiFi if disconnected.
 - Handles CAN errors, SD card errors, and tries to recover.
 - Ensures stable long-term operation.
-

★ 8. Supports Firmware Configurations

- Loads config details (machine type, model, VIN, etc.)
- Sends that info to the dashboard for identification.